P12557/

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04Q 3/00	A2	**(11) International Publication Number:** WO 00/11886 **(43) International Publication Date:** 2 March 2000 (02.03.00)

(21) International Application Number: PCT/SE99/01445

(22) International Filing Date: 24 August 1999 (24.08.99)

(30) Priority Data:

9802840-0	25 August 1998 (25.08.98)	SE
9902683-3	13 July 1999 (13.07.99)	SE

(71) Applicant: TELIA AB (publ) [SE/SE]; Marbackagatan 11, S-123 86 Farsta (SE).

(72) Inventors: BJÖRKMAN, Nils; Fatburs Kvarngata 7, S-118 64 Stockholm (SE). CSEIFENYI, Istvan; Tuillingamasgata 317,1, S-136 63 Haninge (SE). ELLDIN, Jonas; Ekehjelmstorget 4,2, S-118 54 Stockholm (SE). GISGÅRD, Christer; Ösmogatan 7, S-122 48 Enskede (SE). LATOUR-HENNER, Alexander; Orminingeringen 55 C, S-132 33 Saltsjö-Boo (SE). LUNDBERG, Torbjörn; Finnbergsvägen 18, S-131 31 Nacka (SE). ÖSUND, Stefan; Envägen 15, S-138 35 Älta (SE).

(74) Agent: PRAGSTEN, Rolf; Telia Research AB, Vitsandsgatan 9, S-123 86 Farsta (SE).

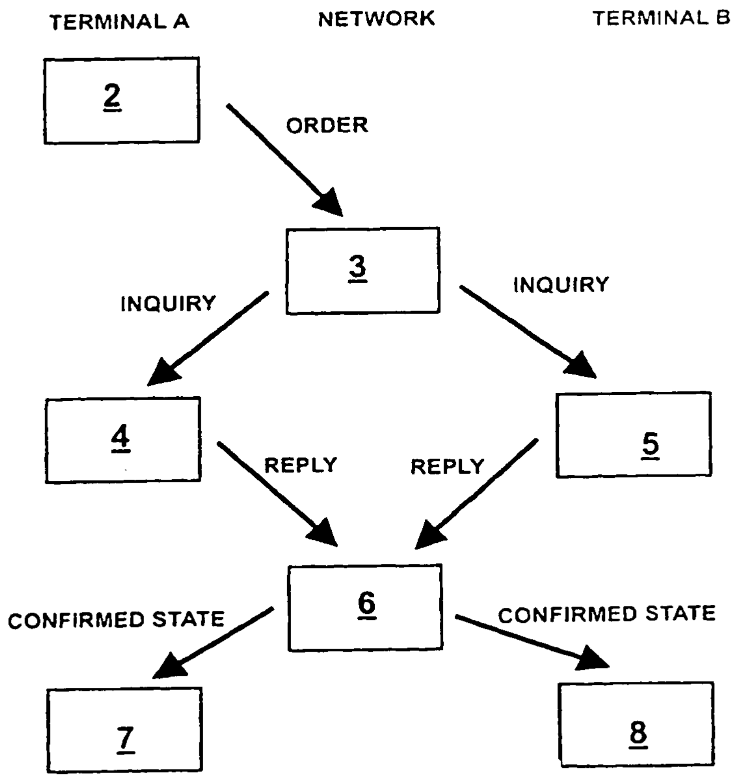
(81) Designated States: EE, LT, LV, NO, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: IMPROVEMENTS IN, OR RELATING TO, TELESERVICE MANAGEMENT SYSTEMS

(57) Abstract

A teleservice management system is adapted to support the provision of a plurality of complex teleservices and has a plurality of intercommunicating subsystems. The teleservice management system negotiates and settles agreements between participants to a service session, resource control architectures within terminals and a resource control architecture in a transmission network. The teleservice management system employs a teleservice control protocol for transmitting messages between said subsystems and thereby controlling a teleservice. The teleservice control protocol is arranged to link a network resource manager, service users and terminal resource managers, into a teleservice control process for facilitating delivery of a teleservice which is optimal, in terms of resource usage, and agreed by a service user. The teleservice control protocol is adapted for use with a plurality of different teleservices including: multiparty, multimedia conference services; tele-game services; tele-shopping services; and tele-education services. The teleservice control protocol messages refer to specific parts of an object oriented model of a teleservice session. The teleservice control protocol includes messages exchanged between: a terminal part and a network part of the teleservice management system; the teleservice management system and a service control graphical user interface; the teleservice management system and an application launcher daemon; and the teleservice management system and network resource managers.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

Improvements in, or Relating to, Teleservice Management Systems

The present invention relates to a teleservice management system, a service platform for managing a plurality of complex teleservices, a telecommunications system including a teleservice management system and a method of managing a plurality of complex teleservices

The first worldwide telecommunication system, i.e. the digital telephone network, has been working for years. The allocation of resources with guaranteed service quality requirements, such as low delay and delay variation, for this single-service system was also solved a long time ago. Resource management is much more complicated in the case of broadband integrated services networks which support several multimedia, multi-connection, multi-party services with different requirements.

In the case of normal telephony, the source terminal is allocated first, after lifting the phone-receiver, then the network resources are allocated as the CCSS #7 signalling message proceeds through the network and the destination terminal is finally allocated. If the called user answers within a given waiting time, the call is established. Otherwise the allocated resources are released in a reverse order. This Bearer Service Layer protocol does not require management functions in the Teleservice Layer.

Because of the complexity and diversity of broadband teleservices, the management functions have to be divided between the Service Management System and the Control Architecture. Since signalling messages in the Teleservice Layer are not yet supported by the current standards, a new protocol has to be designed. This can be considered as third party, overlay signalling, since it is up to the SMS to translate the Teleservice Layer messages to proper bearer signalling messages during call establishment, operation and release. Basically, two type of Bearer Service Layer protocols can be distinguished, the forward and backward allocation, depending on the direction of resource reservation between the source and destination terminals.

In our co-pending patent application, Telia Case 698 (Application No. ....), the content of which is imported herein by reference, there is described a teleservice management system which solves the following problems:

- the TSMS relieves the user from having to provide elaborate and comprehensive technical descriptions in order to invoke the service and to indicate service quality aspects;
- the SU cannot usually express his QoS preference, since the application does not ask him, therefore, the TSMS has an own Service Control interface for interaction with the user;
- allocation, and release, of telecommunication applications and resources in terminals and network;
- co-ordination and integration of telecommunication applications and resources in terminals and network; and
- controlling of resources from the service point of view, thereby maintaining the required quality of service, e.g. adapting resources, such as, communication bearer capabilities, according to different, or shifting, circumstances.

This TSMS provides a general scheme for constructing new teleservices which facilitates the service definition, in general and, in particular, provides a systematic way of handling the different service options, which might otherwise degenerate into an unmanageably large set of possible service variants.

The invention described and claimed in our co-pending application Telia Case 698 provides a teleservice management system adapted to manage a plurality of complex teleservices. This teleservice management system includes a service control module arranged to provide a user with a graphical interface adapted to enable the user to provide data on a required QoS and other parameters, relating to a teleservice which the user wishes to invoke. Means are also provided for storing information relating to service definitions. The teleservice management unit creates an object oriented teleservice model from a request for service, received from said graphical interface via said service control module, using a default mapping between teleservice layer entities and bearer service layer entities.

The present invention relates, inter alia, to a signalling protocol for transmission of control messages in a teleservice system of the type disclosed in our co-pending patent application Telia Case 698.

Two further related inventions are described in our co-pending patent applications, listed below, the content of the specifications of these patent applications is incorporated into this patent specification by reference. The co-pending patent applications are:

- Telia Case 700 (our co-pending patent application ....), which relates to a procedure for creating reservation graphs for the resources in terminals and networks using the teleservice description given by the service provider, said procedure being aimed at producing the most complete teleservice configuration consistent with service specific rules; and

- Telia Case 701 (our co-pending patent application ....), which relates to a service control graphical interface for facilitating communication between a Teleservice Management System and a system user.

The TCP of the present invention is unique. Many protocols exist for directly controlling the bearer network resources, e.g. establishing connections with given parameters. However, the TCP of the present invention controls the teleservice itself, i.e. it operates at a higher level and does not act directly on the resources.

The difference between the TCP of the present invention and other teleservice level protocols is that the TCP herein described involves not only the network resource manager and the user application, but also brings the service user and the terminal resource manager into the control process. This results in a teleservice session which is optimal from the resources' perspective and which is also acknowledged by service users.

The type of the TCP's protocol primitives, and their order, are unique.

The vast majority of teleservice control protocols are applicable to only one, or at most a few, teleservices, while the TCP of the present invention has general validity.

The TCP messages used in the present invention refer to specific parts of an object oriented model of the teleservice session.

According to a first aspect of the present invention, there is provided a teleservice management system, adapted to support the provision of a plurality of complex teleservices and including a plurality of intercommunicating subsystems, characterised in that said teleservice management system includes negotiation means for settling agreements between participants to a session, resource control architectures within terminals and a resource control architecture in a transmission network, said negotiation means including a teleservice control protocol for transmitting messages between said subsystems and thereby controlling a teleservice.

Said teleservice control protocol may be arranged to link a network resource manager, service users and terminal resource managers, into a teleservice control process for facilitating delivery of a teleservice which is optimal, in terms of resource usage, and agreed by a service user.

Said teleservice control protocol may be adapted for use with a plurality of different teleservices.

Said plurality of different teleservices may include:

- multiparty, multimedia conference services;

- tele-game services;

- tele-shopping services; and

- tele-education services.

Said teleservice control protocol messages may refer to specific parts of an object oriented model of a teleservice session.

Said teleservice control protocol may be implemented in a teleservice layer and is transparently transferred in underlying layers.

Said underlying layers may include network layers and transport layers.

Said teleservice control protocol may include messages exchanged between:

- a terminal part and a network part of the teleservice management system;

- the teleservice management system and a service control graphical user interface;

- the teleservice management system and an application launcher daemon; and

- the teleservice management system and network resource managers.

Said teleservice control protocol may have four primitives for communication within said teleservice management system.

Said teleservice control protocol may have four primitives for communication with:

- a service control;
- a terminal resource manager;
- a network resource manager;
- and an application daemon.

Said teleservice management system may have the following interfaces:

- a first interface, between a terminal part and a signalling emulator;
- a second interface, between a terminal and a service control;
- a third interface, between a terminal and an application launcher daemon;
- a fourth interface, between a terminal and a terminal resource manager; and
- a fifth interface, between a signalling emulator and a network resource manager.

Said teleservice control protocol may be adapted to transfer information relating to a part of a teleservice object model through said first interface.

Said teleservice control protocol may be adapted to transfer information relating to a part of a teleservice object model through said second interface.

Said teleservice control protocol may be adapted to transfer information relating to a name and parameters, associated with an application employed by a teleservice, through said third interface.

Said teleservice control protocol may be adapted to transfer information relating to a specification of terminal resources, required by a teleservice, through said fourth interface.

Said teleservice control protocol may be adapted to transfer information relating to a specification of network resources, required by a teleservice, through said fifth interface.

According to a second aspect of the present invention, there is provided a telecommunications system, logically split between an application layer, a teleservice layer, a bearer service layer and a resource layer, characterised in that said teleservice layer includes a teleservice management system as set forth in any previous paragraph.

According to a third aspect of the present invention, there is provided a service platform adapted to support the provision of a plurality of complex teleservices and including a plurality of intercommunicating subsystems, terminals and a telecommunications network, characterised in that said service platform includes resource management means, and in that a teleservice control protocol is employed for implementing protocol agents both in said terminals and in said network.

Protocol agents may be implemented in said terminals by utilising a resource control application provider interface, forming part of a network control architecture, as an interface between network resource management and a signal emulator.

Said terminals may have at least one interface towards a user.

Said terminals may have an interface towards an application.

Said terminals may have an interface towards a terminal resource manager.

Said teleservice control protocol may be adapted for use with a plurality of different teleservices.

Said plurality of different teleservices may include:

- multiparty, multimedia conference services;

- tele-game services;

- tele-shopping services; and

- tele-education services.

Said teleservice control protocol messages may refer to specific parts of an object oriented model of a teleservice session.

Said teleservice control protocol may have four primitives for communication within said teleservice management system.

According to a fourth aspect of the present invention, there is provided a method of managing a plurality of complex teleservices employing a teleservice management system including a plurality of intercommunicating subsystems, characterised by settling agreements between participants to a session by exchanging messages using a teleservice control protocol, said teleservice control protocol linking a network resource manager, service users and terminal resource managers into a teleservice control process for facilitating delivery of a teleservice which is optimal in terms of resource usage and agreed by a service user.

Said teleservice control protocol may operate with a plurality of different teleservices.

Said plurality of different teleservices may include:

- muitiparty, multimedia conference services;

- tele-game services;

- tele-shopping services; and

- tele-education services.

Said teleservice control protocol messages may refer to specific parts of an object oriented model of a teleservice session.

Said teleservice control protocol may be implemented in a teleservice layer and transparently transfer teleservice protocol messages in underlying layers.

Said underlying layers may include network layers and transport layers.

Said teleservice control protocol may include messages exchanged between:

- a terminal part and a network part of the teleservice management system;
- the teleservice management system and a service control graphical user interface;
- the teleservice management system and an application launcher daemon; and
- the teleservice management system and network resource managers.

Said teleservice control protocol may have four primitives for communication within said teleservice management system.

Said teleservice control protocol may have four primitives for communicating with:

- a service control;
- a terminal resource manager;
- a network resource manager;

- and an application daemon.

Said teleservice management system may have the following interfaces:

- a first interface, between a terminal part and a signalling emulator;
- a second interface, between a terminal and a service control;
- a third interface, between a terminal and an application launcher daemon;
- a fourth interface, between a terminal and a terminal resource manager; and
- a fifth interface, between a signalling emulator and a network resource manager.

Said teleservice control protocol may be used to transfer information relating to a part of a teleservice object model through said first interface.

Said teleservice control protocol may be used to transfer information relating to a part of a teleservice object model through said second interface.

Said teleservice control protocol may be used to transfer information relating to a name and parameters, associated with an application employed by a teleservice, through said third interface.

Said teleservice control protocol may be used to transfer information relating to a specification of terminal resources, required by a teleservice, through said fourth interface.

Said teleservice control protocol may be used to transfer information relating to a specification of network resources, required by a teleservice, through said fifth interface.

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 illustrates the main messages of the teleservice control protocol of the present invention.

Figures 2 to 8 show the blocks 2 to 8, respectively, of Figure 1, in greater detail.

Figure 9 illustrates the environment of the teleservice control protocol of the present invention.

Figure 10 illustrates the use of a mux for sharing a TSP/SIC signalling channel.

Figure 11 illustrates mapping between messages.

Figure 12 illustrates the process of implementing different operations on different objects in the same message.

Figure 13 illustrates the process of modifying a LVSI.

Figure 14 illustrates the process of rolling back a LVSI.

Figure 15 illustrates the process of object deletion in a LVSI.

Figure 16 illustrates the process by which the Service Control in a terminal handles service dependent time-outs.

Figure 17 illustrates the process of sending Inquiries to parties who have not sent a Reply.

Figure 18 shows the contents of a Reply.

Figure 19 illustrates the basic flow for setting up a simple bearer service.

Figure 20 illustrates the basic flow for terminating a simple bearer service.

Figure 21 illustrates how states in an object in the terminal and the corresponding object in the network change when messages are passed between them.

Figure 22 illustrates the use of id.

Table 1 lists objects of the Telecommunication Service Description Model.

Table 2 lists typical messages for establishing a teleservice session between two parties.

Table 3 contains the structure of a TSP Message.

Table 4 lists attributes in the class TSP_MSG.

Table 5 lists a protocol Descriptor for a TSP Message.

Table 6 contains the structure of a PDU for a TSP/SIC.

Table 7 lists attributes in the class SIC_MSG.

Table 8 is a table of message types.

Table 9 contains the structure of a PDU for a USM Leg.

Table 10 lists attributes in the class USM_LEG.

Table 11 contains the structure of a PDU for a PARTY_LEG.

Table 12 lists attributes in the class PART_LEG.

Table 13 lists attributes in the class ELEMENT_HEAD_IN_LEG.

Table 14 is a table of Element Types.

Table 15 is a table of Operations.

Table 16 is a table of Responses.

Table 17 is a table of Causes.

Table 18 contains the structure of a PDU for a SI_HEAD.

Table 19 lists attributes in the class SI_HEAD.

Table 20 contains the structure of a PDU for the USM_ELEMENT.

Table 21 lists attributes in the class USM_ELEMENT.

Table 22 is a table of presence in USM.

Table 23 contains the structure of a PDU for an ASM_ELEMENT.

Table 24 lists attributes in the class ASM_ELEMENT.

Table 25 is a table of presence in ASM.

Table 26 contains the structure of a PDU for a PARTY_ELEMENT.

Table 27 lists attributes in the class PARTY_ELEMENT.

Table 28 is a table of presence in a PARTY_ELEMENT.

Table 29 contains the structure of a PDU for PARTY_ID.

Table 30 lists attributes in the class PARTY_ID.

Table 31 contains the structure of a PDU for a PE_ELEMENT.

Table 32 lists attributes in the class PE_ELEMENT.

Table: 33 is a table of presence in a PE_ELEMENT.

Table 34 contains the structure of a PDU for a PAE_ELEMENT.

Table 35 lists attributes in the class PAE_ELEMENT.

Table 36 is a table of presence in a PAE_ELEMENT.

Table 37 is a table for direction in a PAE_ELEMENT.

Table 38 contains the structure of a PDU for a SM_ELEMENT.

Table 39 lists attributes in the class SM_ELEMENT.

Table 40 contains the structure of a PDU for an ACE_ELEMENT.

Table 41 list attributes in the class ACE_ELEMENT.

Table 42 contains the structure of a PDU for class TRAFFIC_DESCRIPTOR.

Table 43 lists attributes in the class TRAFFIC_DESCRIPTOR.

Table 44 is a table for type in the class TRAFFIC_DESCRIPTOR.

Table 45 gives the meaning of the attributes Operation, Response and Cause in an Order.

Table 46 gives the meaning of the attributes Operation, Response and Cause in an Inquiry.

Table 47 gives the meaning of the attributes Operation, Response and Cause in a Reply.

Table 48 gives the meaning of the attributes Operation, Response and Cause in a CS.

In order to assist the reader to a better understanding of this patent specification, a glossary of some the abbreviations used herein is set out below:

ABB	Application Building Block
ACE	?
AP	Application Provider
API	?
APLD	Application Launcher Daemon
ASM	Abstract Service Module
ATM	Asynchronous Transfer Mode
CAC	Call Admission Control
EMMA	Experimental Middleware for ATM
GSMP	?
IAPI	IP/ATM Protocol Interface
IP	Internet Protocol
LVSI	Local View of Service Instance
MoD	Movie on Demand
NAP	Network Access Program

NAPI	Native ATM Protocol Interface
NIC	?
NP	Network Provider
NRM	Network Resource Manager
PAE	Party ASM Edge
PE	Party Edge
QoS	Quality of Service
RR	?
SC	Service Control
SI	Service Instance
SIC	Service Instance Control
SIGNE	Signalling Emulator
SM	Service Module
SMS	Service Management System
SU	Service User
TCSD	Telecommunication Service Description
TP	Terminal Profile
TRM	Terminal Resource Manager

TCP	TeleService Control Protocol
TSMS	TeleService Management System
TSP	Telecommunication Service Protocol
UPC	Usage Parameter Control
UQL	User Quality Level
USM	User Service Module

Control of distributed multimedia services requires internal communication between the components of a teleservice management system (TSMS), employed to control provision of these services. Internal communication is needed in order to:

- settle agreements between those participating in a session and the resources employed in the provision of the service;
- control architectures within terminals accessing a service and the network providing transmission functions for the service;
- control the dynamic behaviour of a teleservice session;
- update the state models of the teleservice session, stored in the terminals and in the network.

The present invention relates to a teleservice management system employing a teleservice control protocol which implements these functions.

The Teleservice Control Protocol (TCP) is used by the sub-systems of the TSMS, of the present invention, for implementing the functions set out above. The TCP includes messages exchanged between:

- the terminal part (EMMA) and network part (SIGNE) of the TSMS;

- the TSMS and the Service Control graphical user interface;

- the TSMS and the Application Launcher Daemon; and

- the TSMS and the Terminal and Network Resource Managers.

It implies four primitives for communication within the TSMS, see Figures 1 to 8, and four primitives for communication with:

- the SC;

- the Terminal Resource Manager (TRM);

- Network Resource Manager (NRM); and

- the Application Launcher Daemon (APLD).

It should be noted that the blocks labelled 2 to 8 in Figure 1 are illustrated in greater detail in Figures 2 to 8, respectively.

The typical order of messages and description of the primitives are given in Table 2. Table 1 lists the different information transferred through the different interfaces of the TSMS.

The creation, modification and termination, of teleservice sessions proceeds using the same message types, but with different contents. There are several possible orders for TCP messages, depending on the availability of terminal and network resources and the answers of service users and applications. A typical example is given in Table 2, where the notation "10a" and "10b" means that these steps may happen simultaneously, or in any other order in time.

The TCP is implemented in the Teleservice Layer and is, therefore, transparently transferred in the underlying layers, e.g. network and transport layers.

The TCP, of the present invention, can be applied within a Telecommunication Service Management System. Due to the general nature of the teleservice modelling approach, the same protocol primitives can be applied to any teleservice, e.g. multiparty multimedia conference, tele-games, tele-shopping, tele-education, etc..

Moreover, the TCP can also be used in Service Platforms as an extension for Resource Management. In this case, protocol agents should be implemented both in the network and in the terminals. The former utilise the resource control API of the network control architecture, e.g. GSMP, as the NRM/SIGNE interface. The latter should have at least one interface toward the user and, optionally, two more interfaces toward the application and the terminal resource manager. However, some of the functionality of TCP can be kept even if only the NRM and SC are involved in a negotiation.

The invention can be placed on the top of a general network resource control architecture, so called switchlet, which, in this case, replaces the network resource manager.

The operating environment of the present invention is illustrated in Figure 9, which is self explanatory and will be readily understood by those skilled in the art.

TSP is a family of protocols which sit on top of TCP/IP for Signalling in SIGNE 2. The family includes TSP/HTTP, TSP/REG, TSP/TCSD, TSP/DIR, TSP/SIC, TSP/MUX. The protocol will now be described, in detail, from the point of view of implementation on the terminal side of TSP/SIC. The way in which SIGNE 2 is implemented on the network side of TSP/SIC is not described.

TSP/SIC is used for distribution of objects in the Service Instance to terminals. The objects in LVSI have one part that is local and another that is the distributed part of the Service Instance in the network. Only attributes and services needed in the terminal's local view are in LVSI. The local part comprises attributes and services for communicating with the ABBs and the Service Control.

The basic flow is as follows:

1. Order from the initiating terminal to the network.

2. Inquiry from the network to the terminal.

3. Reply from the terminal.

4. Confirmed State from the network to the terminal

The terminal sends an order to the network containing the user's request for a Service Instance. The network processes the order and sends out Inquiries to all parties to the service.

The Inquiry can be a request for the whole Service Instance, or just a part of it. The Inquiry specifies what resources the terminal needs to allocate to participate in the inquired part of the service. Whether the user, at the terminal, needs to be questioned about the inquiry, or not, is service dependent. If needed, the Service Control may modify both the question to the user and the answer from the user, before sending a reply to the network.

When the terminal has processed the Inquiry and allocated ABBs, it gives an answer by sending a Reply to the network. In the Reply, objects from the Inquiry are returned with a status of accept, or reject. The network processes the Inquiry and when the demands for setting up a SI is True, the network sends a Confirmed State to all those who sent a Reply. The Confirmed State tells the terminal about the Confirmed State of the SI. For all objects that are active this mean that a connection is established and the ABBs can start sending and receiving . The network waits for late Replies and sends a new Confirmed State when they are received.

Fig 19 shows a basic flow for setting up a simple bearer service using 2 parties, one USM and one ASM.

Fig 20 shows a basic flow for terminating a simple bearer service.

When the SI is terminated an Inquiry is needed to tell the ABBs to finish the current session and stop sending. In this case the Confirmed State just indicates that termination has been completed in the network. The terminal is not allowed to throw away a LVSI before a Confirmed State says so. For ABBs that do not require precise terminations, the terminal (Service Control) may send a Reply before killing the ABB. Some types of ABB do not need to be killed, e.g. hardware video decoders.

Figure 21 illustrates the way in which states in an object in the terminal and the corresponding object in the network change when messages are past between them. The example, illustrated by Figure 21, shows the successful creation of an object.

The Order message contains Message head, SI_HEAD, USM legs, and the service part of the Party legs (not SMs and ACEs).

The Inquiry message contains Message head, SI_HEAD, USM legs, the whole Party leg for the receiver of the message and the service part of the other Party legs.

The Reply message contains Message head, SI_HEAD and the sending Party's whole Party leg. The USM leg contains only size_of_usm_leg and no objects.

The CS message contains Message head, SI_HEAD, USM legs, the whole Party leg for the receiver of the message and the service part of the other Party legs.

The purpose of the messages is to update the other side's service instance. It contains the attributes from the objects in the SI, and in the LVSI, that are needed in the other end.

A SIC message holding one buffer for USM legs and one for Party legs is put into a TSP message. In the USM leg and the PARTY leg the sequence is important, it provides information about relations downwards of the legs for example, which USM an ASM belong to, which Party a PE belong to, and so on.

The structure of messages will now be described.

Sizes of types: char = 1 byte, int = 4 bytes. For enums an int is used. In enum types the first value is always zero and stands for Undefined. It is the recommended default value when the object is created. Unused bytes are set to Zero.

The size of TSP_MSG, USM_LEG and PARTY_LEG is variable.

All attributes in the classes are mandatory. Some classes are mandatory in all messages like: TSP_MSG, SIC_MSG, SI_HEAD, USM_LEG, Party_LEG.

A TSP message is built in an object of class TSP_MSG, see Table 3. Table 4 shows the attributes in class TSP_MSG, Table 5 is the table for protocol descriptors.

The TSP_MSG is mandatory in all TSP messages.

The dest_id and org_id will now be considered.

Dest_id is the identity of the destination to which messages are sent. In the network it is a reference to the SI, in Emma it is a reference to the right LVSI. If the receiving instance doesn't exist then dest_id = 0 and the receiver knows that this instance has yet to be created. Dest_id=0 is only valid in the first Order creating the SI and in Inquiries to the B-sides.

Org_id is a reference to the address to which messages are to be sent back for this Instance.

The value 0 for a Org_id is not valid.

Consider the following example:

```
Order
    dest_id = 0         # Teliing this is a new SI.
    org_id = 73         # Telling where you want the Inquiry to be
                         # returned to.
Inquiry to calling
    dest_id = 73        # Network sends back to the right LVSI.
    org_id = 112        # Network tells the terminal where it wants
                        # the Reply.
Inquiry to called
    dest_id = 0         # Network asks the other terminal to create a LVSI.
    org_id = 114        # Network tells the terminal where it wants
                        # the Reply.
Reply from calling
    dest_id = 112       # Terminal sends back to the right SI.
    org_id = 73         # Terminal tells the network where it wants
                        # the CS.
Reply from called
    dest_id = 114       # Terminal sends back to the right SI.
    org_id = 19         # Terminal tells the network where it wants
                        # the CS.
```

```
CS to calling
    dest_id = 73        # Network sends back to the right LVSI.
    org_id = 112        # Network tells the terminal where it wants
                        # new Orders.
CS to called
    dest_id = 19        # Network asks the other terminal to create a LVSI.
    org_id = 114        # Network tells the terminal where it wants
                        # new Orders.
```

The structure of the PDU for TSP/SIC is of class SIC_MSG and is given in Table 6. The attributes in class SIC_MSG are given in Table 7, Table 8 is the table for Message Type.

The SIC_MSG is mandatory in all TSP/SIC messages.

The structure of the PDU for the USM leg is of class USM_LEG and is given in Table 9. The attributes in class USM_LEG are given in Table 10.

The USM_LEG is mandatory in all messages, but there is no requirement for there to be any elements in it. In this case, size_of_usm_leg = sizeof(int) = 4.

In the USM_LEG the relationship upwards between objects is shown by putting related ASMs after its USM. ASMs under an USM is added in sequence after the USM and before the next USM.

The structure of the PDU for PARTY leg is of class PARTY_LEG and is given in Table 11. The attributes in class PARTY_LEG are set out in Table 12.

The PARTY_LEG is mandatory in all messages, but there is no requirement for there to be any elements in it. In this case size_of_party_leg = sizeof(int) = 4.

In the Party leg, the relationship upwards, between objects, is indicated by putting a related PAE after its PE and so on.

The classes for SI_HEAD, USM, ASM, PARTY, PE, PAE, SM and ACE all use the class ELEMENT_HEAD_IN_LEG for common parts for all objects in the legs. Table 13 gives the attributes in class ELEMENT_HEAD_IN_LEG:

The terminal_id is used to identify an object sent in an order when it returns in an inquiry. When an object is created in SI, the network_id is used as reference between LVSI and SI. The terminal_id is not be used after the object has obtained the network_id. Some objects like SM and ACE are not given any terminal_id because they are not in the Order. The terminal_id is not unique in the SI because different terminals can use the same terminal_id values for different objects.

Only the party who has initiated, i.e. ordered, the operation on the object will receive the terminal_id set. Other parties will receive terminal_id = 0. This can be used when the terminal checks if the inquiry is a result of an own order.

A terminal doesn´t know the party_id of the initiator of an Order but it does know if it is an operation which the terminal initiated itself.

Fig 22 illustrates an example of the use of id.

A Table for Element Type is set out in Table 14, a Table for Operations is set out in Table 15, a Table for Responses is set out in Table 16 and a Table for Causes, is set out in Table 17

The structure of the PDU for a SI_HEAD is set out in Table 18 and Table 19 list the attributes in the class SI_HEAD.

The SI_HEAD is used for operations on the SI and LVSI object in the SI and LVSI. For example to remove a whole SI.

Service_Utility_ref is a reference to choose a Service Control in the Terminal. This mapping is stored locally in the Terminal.

TCS_ref a reference to the TCSD to be used. In TSP/TCSD the attribute for identifying a TCSD is TCSD_HEAD::TCSD_id. Several versions of the same TCSD are possible, therefore an attribute for version is needed, TCS_version.

The structure of the PDU for USM_ELEMENT is given in Table 20 and the attributes in class USM_ELEMENT are listed in Table 21:

The presence should be set in the Order and not altered in the Reply, Table 22 is the table for presence in USM.

The structure for the PDU for an ASM_ELEMENT is given in Table 23 and the attributes in the class ASM_ELEMENT are listed in Table 24.

The presence should be set in the Order and not altered in the Reply, Table 25 is the table for presence in ASM:

The structure of the PDU for a PARTY_ELEMENT is given in Table 26 and the attributes in class PARTY_ELEMENT are listed in Table 27.

The presence should be set in the Order and not altered in the Reply, Table 28 gives the table for presence in PARTY_ELEMENT.

The structure of the PDU for a PARTY_ID is given inTable 29 and the attributes in the class PARTY_ID are listed in Table 30.

The structure of the PDU for PE_ELEMENT is given inTable 31 and the attributes in class PE_ELEMENT are listed in Table 32.

The presence should be set in the Order and not altered in the Reply, Table

33 gives the table for presence in PE_ELEMENT.

The structure of the PDU for a PAE_ELEMENT is given in Table 34 and the attributes in the class PAE_ELEMENT are listed in Table 35.

The presence should be set in the Order and not altered in the Reply, Table 36 gives the table for presence in PAE_ELEMENT and Table 37 is the table for direction in the PAE_ELEMENT.

The structure of the PDU for SM_ELEMENT is illustrated in Table 38 and the attributes in class SM_ELEMENT are listed in Table 39.

The structure of the PDU for the ACE_ELEMENT is given in Table 40 and the attributes in the class ACE_ELEMENT are listed in Table 41.

The structure of the PDU for the class TRAFFIC_DESCRIPTOR is given in Table 42, the attributes in class TRAFFIC_DESCRIPTOR are listed in Table 43 and Table 44 is the table for Type in the class TRAFFIC_DESCRIPTOR.

Attributes for changing state in TSP/SIC are set out below.

Attribute in the message head:

message type: { order, inquiry, reply, CS }

Attributes in the other objects:

operation : { none, create, modify, remove }

response: { none, accept, reject }

cause: { interger value (0 = No cause, 1-max int = Causes) }

The response attribute is only valid for the latest message, cause is used when the response == reject.

To create an object in the other model :

message type = order

and for the object

operation = create

response = none

cause = No cause

When the network asks the terminal:

message type = inquiry

and in the object

operation = create

response = accept

cause = No cause

If the network can not accept something in the order:

message type = inquiry

and in the object

operation = create

response = reject

cause = "Why cause"

When the terminal accepts the inquiry:

message type = reply

and in the object

operation = create

response = accept

cause = No cause

or rejects something in the inquiry

message type = reply

and in the object

operation = create

response = reject

cause = "Why cause"

When the network is responding to a reply a CS is sent:

message type = CS

and in the object

operation = create

response = accept

cause = No cause

In an Order the attributes Response and Cause have no meaning and the attribute Operation has the following meaning:

Operation = None : The object exist and shall not be affected.

Operation = Create : The Object shall be created.

Operation = Modify : Attributes in the object shall be modified in some way.

Ope⁻ation = Remove : The object shall be removed.

Table 45 lists the meaning of the attributes Operation, Response and Cause in an Order, Table 46 lists the meaning of the attributes Operation, Response and Cause in a Inquiry, Table 47 lists the meaning of the attributes Operation, Response and Cause in a Reply and Table 48 lists the meaning of the attributes

Operation, Response and Cause in a CS:

Due to the limited number of open TCP/IP connections in Vxworks, SIGNE will only have one TCP-connection to every terminal for TSP/SIC. In the terminal the NAP, or the NET object, have to implement a mux for sharing the same TSP/SIC signalling channel between all local service instances, see Figure 10. For the other necessary connections, e.g. for registration, http, directory, etc., separate connections can be used.

Mapping between messages is illustrated in Figure 11. The process of implementing different operations on different objects in the same message is illustrated in Figure 12. The process of modifying a LVSI is illustrated in Figure 13. The process of rolling back a LVSI is illustrated in Figure 14. The process of object deletion in a LVSI is illustrated in Figure 15. The process by which the Service Control in a terminal Handles service dependent time-outs is illustrated in Figure 16. The process of sending Inquiries to parties who have not sent a Reply about changes in the SI is illustrated in Figure 17. Finally the contents of a Reply are illustrated in Figure 18. Figures 11 to 18 are self explanatory and will be readily understood by those skilled in the art and will not, therefore, be described in further detail.

CLAIMS

1. A teleservice management system, adapted to support the provision of a plurality of complex teleservices and including a plurality of intercommunicating subsystems, characterised in that said teleservice management system includes negotiation means for settling agreements between participants to a session, resource control architectures within terminals and a resource control architecture in a transmission network, said negotiation means including a teleservice control protocol for transmitting messages between said subsystems and thereby controlling a teleservice.

2. A teleservice management system, as claimed in claim 1, characterised in that said teleservice control protocol is arranged to link a network resource manager, service users and terminal resource managers, into a teleservice control process for facilitating delivery of a teleservice which is optimal, in terms of resource usage, and agreed by a service user.

3. A teleservice management system, as claimed in either claim 1, or claim 2, characterised in that said teleservice control protocol is adapted for use with a plurality of different teleservices.

4. A teleservice management system, as claimed in claim 3, characterised in that said plurality of different teleservices include:

- multiparty, multimedia conference services;
- tele-game services;
- tele-shopping services; and
- tele-education services.

5. A teleservice management system, as claimed in any previous claim, characterised in that said teleservice control protocol messages refer to specific parts of an object oriented model of a teleservice session.

6. A teleservice management system, as claimed in any previous claim, characterised in that said teleservice control protocol is implemented in a teleservice layer and is transparently transferred in underlying layers.

7. A teleservice management system, as claimed in claim 6, characterised in that said underlying layers include network layers and transport layers.

8. A teleservice management system, as claimed in any previous claim, characterised in that said teleservice control protocol includes messages exchanged between:

- a terminal part and a network part of the teleservice management system;

- the teleservice management system and a service control graphical user interface;

- the teleservice management system and an application launcher daemon; and

- the teleservice management system and network resource managers.

9. A teleservice management system, as claimed in any previous claim, characterised in that said teleservice control protocol has four primitives for communication within said teleservice management system.

10. A teleservice management system, as claimed in any previous claim,

characterised in that said teleservice control protocol has four primitives for communication with:

- a service control;
- a terminal resource manager;
- a network resource manager;
- and an application daemon.

11. A teleservice management system, as claimed in any previous claim, characterised in that said teleservice management system has the following interfaces:

- a first interface, between a terminal part and a signalling emulator;
- a second interface, between a terminal and a service control;
- a third interface, between a terminal and an application launcher daemon;
- a fourth interface, between a terminal and a terminal resource manager; and
- a fifth interface, between a signalling emulator and a network resource manager.

12. A teleservice management system, as claimed in claim 11, characterised in that said teleservice control protocol is adapted to transfer information relating to a part of a teleservice object model through said first interface.

13. A teleservice management system, as claimed in either claim 11, or claim 12, characterised in that said teleservice control protocol is adapted to transfer information relating to a part of a teleservice object model through said second interface.

14. A teleservice management system, as claimed in any of claims 11 to 13, characterised in that said teleservice control protocol is adapted to transfer information relating to a name and parameters, associated with an application employed by a teleservice, through said third interface.

15. A teleservice management system, as claimed in any of claims 11 to 14, characterised in that said teleservice control protocol is adapted to transfer information relating to a specification of terminal resources, required by a teleservice, through said fourth interface.

16. A teleservice management system, as claimed in any of claims 11 to 15, characterised in that said teleservice control protocol is adapted to transfer information relating to a specification of network resources, required by a teleservice, through said fifth interface.

17. A service platform adapted to support the provision of a plurality of complex teleservices and including a plurality of intercommunicating subsystems, terminals and a telecommunications network, characterised in that said service platform includes resource management means, and in that a teleservice control protocol is employed for implementing protocol agents both in said terminals and in said network.

18. A service platform, as claimed in claim 17, characterised in that protocol agents are implemented in said terminals by utilising a resource control application provider interface, forming part of a network control architecture, as an interface between network resource management and a signal emulator.

19. A service platform, as claimed in either claim 17, or 18, characterised in that

said terminals have at least one interface towards a user.

20. A service platform, as claimed in claim 19, characterised in that said terminals have an interface towards an application.

21. A service platform, as claimed in either claim 19, or 20, characterised in that said terminals have an interface towards a terminal resource manager.

22. A service platform, as claimed in any of claims 17 to 21, characterised in that said teleservice control protocol is adapted for use with a plurality of different teleservices.

23. A service platform, as claimed in claim 22, characterised in that said plurality of different teleservices include:

- multiparty, multimedia conference services;
- tele-game services;
- tele-shopping services; and
- tele-education services.

24. A service platform, as claimed in any of claims 17 to 23, characterised in that said teleservice control protocol messages refer to specific parts of an object oriented model of a teleservice session.

25. A service platform, as claimed in any of claims 17 to 24, characterised in that said teleservice control protocol has four primitives for communication within said teleservice management system.

26. A telecommunications system, logically split between an application layer,

a teleservice layer, a bearer service layer and a resource layer, characterised in that said teleservice layer includes a teleservice management system as claimed in any of claims 1 to 16.

27. A method of managing a plurality of complex teleservices employing a teleservice management system including a plurality of intercommunicating subsystems, characterised by settling agreements between participants to a session by exchanging messages using a teleservice control protocol, and by said teleservice control protocol linking a network resource manager, service users and terminal resource managers into a teleservice control process for facilitating delivery of a teleservice which is optimal in terms of resource usage and agreed by a service user.

28. A method as claimed in claim 27, characterised by said teleservice control protocol operating with a plurality of different teleservices.

29. A method, as claimed in claim 28, characterised by said plurality of different teleservices including:

- multiparty, multimedia conference services;
- tele-game services;
- tele-shopping services; and
- tele-education services.

30. A method, as claimed in any of claims 27 to 29, characterised by said teleservice control protocol messages referring to specific parts of an object oriented model of a teleservice session.

31. A method, as claimed in any of claims 27 to 30, characterised by

implementing said teleservice control protocol in a teleservice layer and by transparently transferring teleservice protocol messages in underlying layers.

32. A method, as claimed in any of claims 27 to 31, characterised by said underlying layers including network layers and transport layers.

33. A method, as claimed in any of claims 27 to 32, characterised by said teleservice control protocol including messages exchanged between:

- a terminal part and a network part of the teleservice management system;
- the teleservice management system and a service control graphical user interface;
- the teleservice management system and an application launcher daemon; and
- the teleservice management system and network resource managers.

34. A method, as claimed in any of claims 27 to 33, characterised by said teleservice control protocol having four primitives for communication within said teleservice management system.

35. A method, as claimed in any of claims 27 to 34, characterised by said teleservice control protocol having four primitives for communicating with:

- a service control;
- a terminal resource manager;

- a network resource manager;

- and an application daemon.

36. A method, as claimed in any of claims 27 to 35, characterised by said teleservice management system having the following interfaces:

- a first interface, between a terminal part and a signalling emulator;

- a second interface, between a terminal and a service control;

- a third interface, between a terminal and an application launcher daemon;

- a fourth interface, between a terminal and a terminal resource manager; and

- a fifth interface, between a signalling emulator and a network resource manager.

36. A method, as claimed in claim 35, characterised by using said teleservice control protocol to transfer information relating to a part of a teleservice object model through said first interface.

37. A method, as claimed in either claim 35 or 36, characterised by using said teleservice control protocol to transfer information relating to a part of a teleservice object model through said second interface.

38. A method, as claimed in either claim 35 or 37, characterised by using said teleservice control protocol to transfer information relating to a name and parameters, associated with an application employed by a teleservice, through said third interface.

39. A method, as claimed in either claim 35 or 38, characterised by using said teleservice control protocol to transfer information relating to a specification of terminal resources, required by a teleservice, through said fourth interface.

40. A method, as claimed in either claim 35 or 38, characterised by using said teleservice control protocol to transfer information relating to a specification of network resources, required by a teleservice, through said fifth interface.

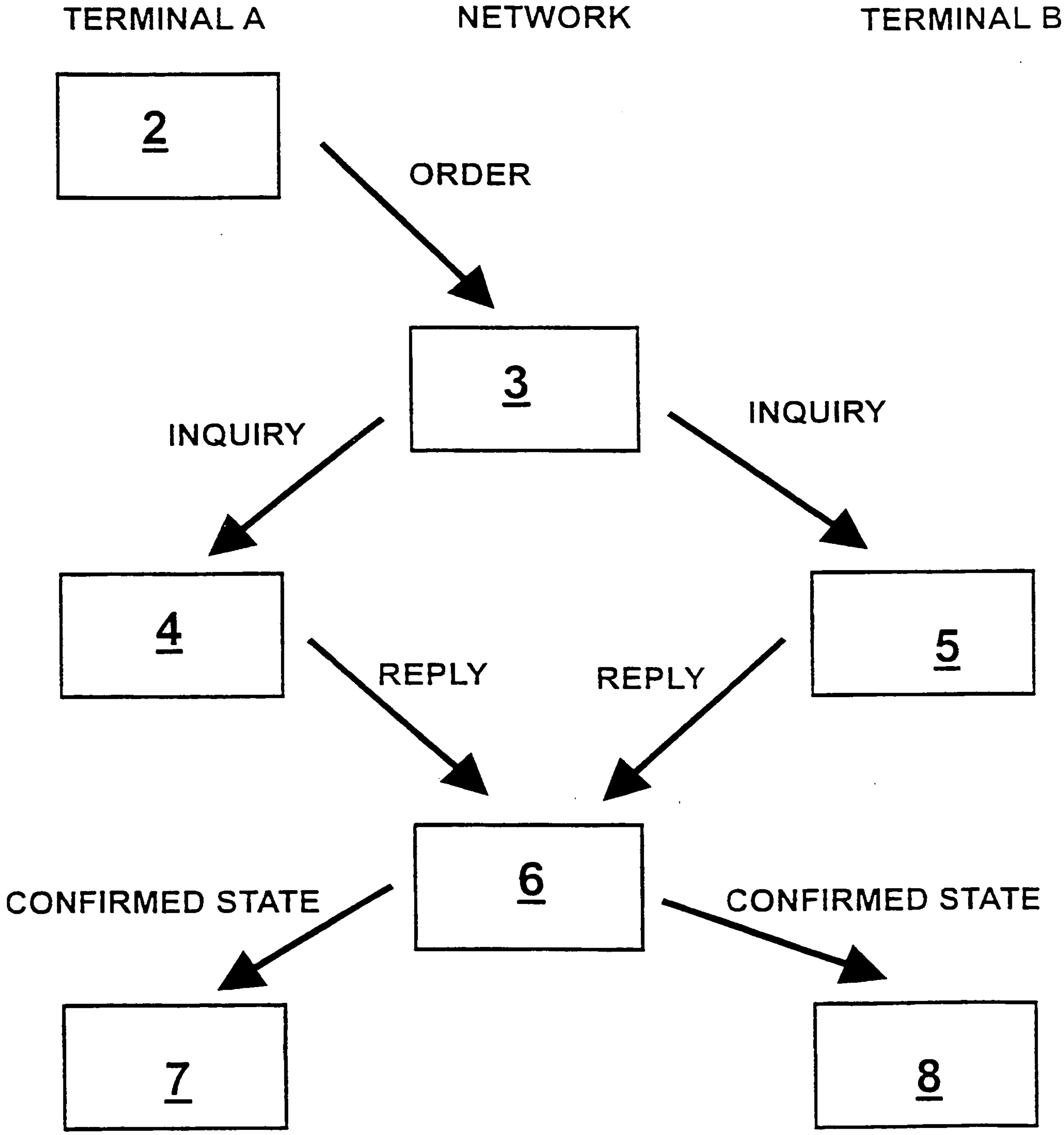

TERMINAL A
NETWORK
TERMINAL B
2
ORDER
3
INQUIRY
INQUIRY
4
5
REPLY
REPLY
6
CONFIRMED STATE
CONFIRMED STATE
7
8

FIGURE 1

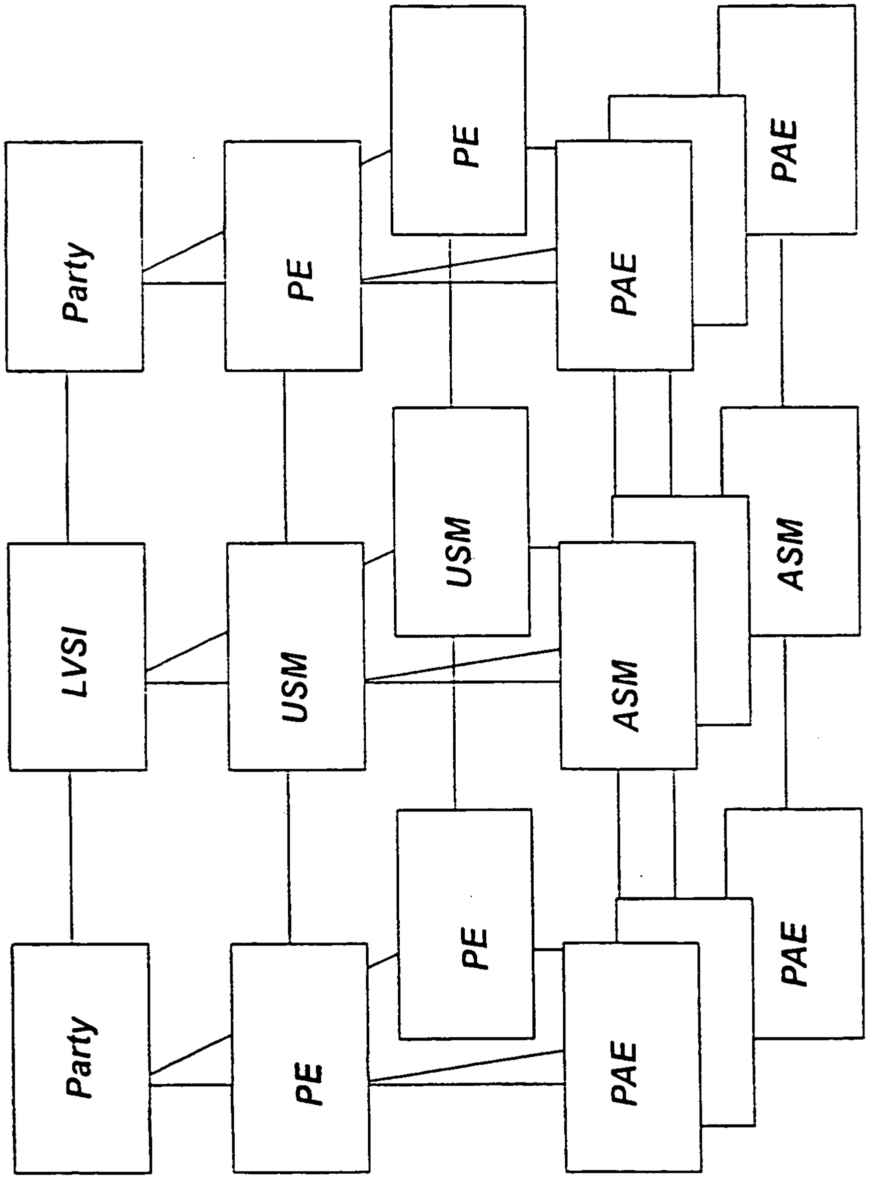


Figure 2

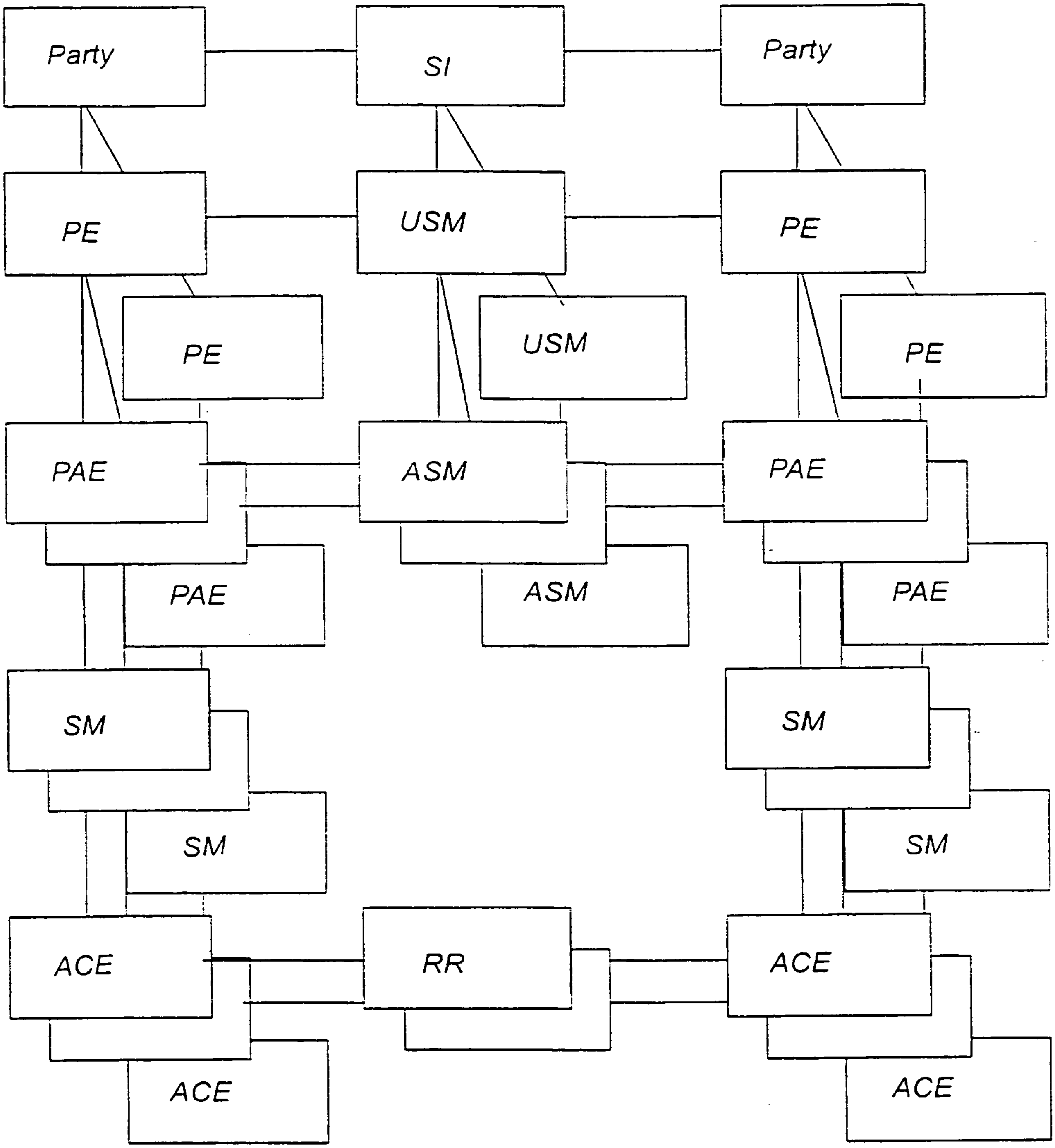

Party
SI
Party
PE
USM
PE
PE
USM
PE
PAE
ASM
PAE
PAE
ASM
PAE
SM
SM
SM
SM
ACE
RR
ACE
ACE
ACE

FIGURE 3

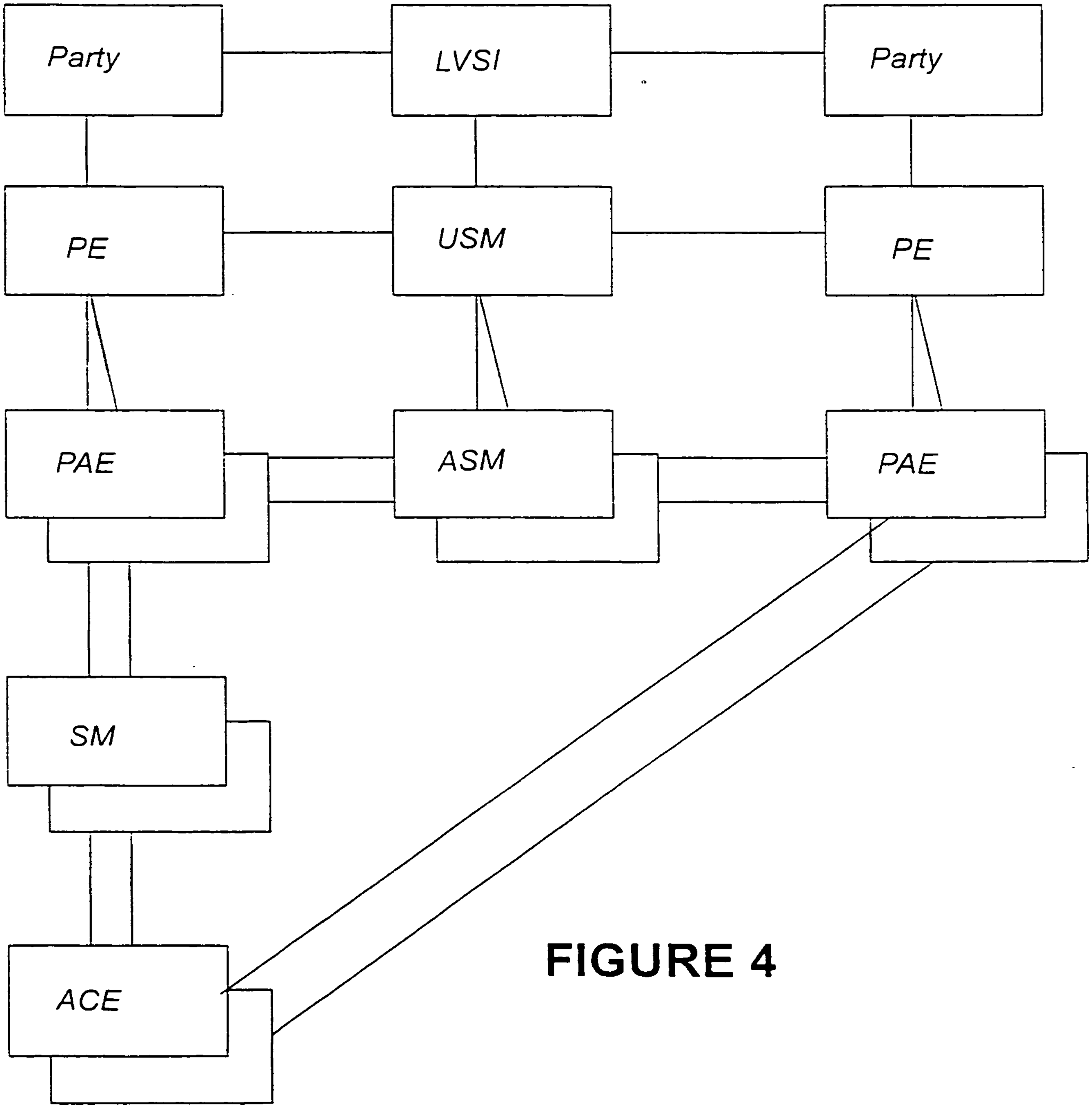


FIGURE 4

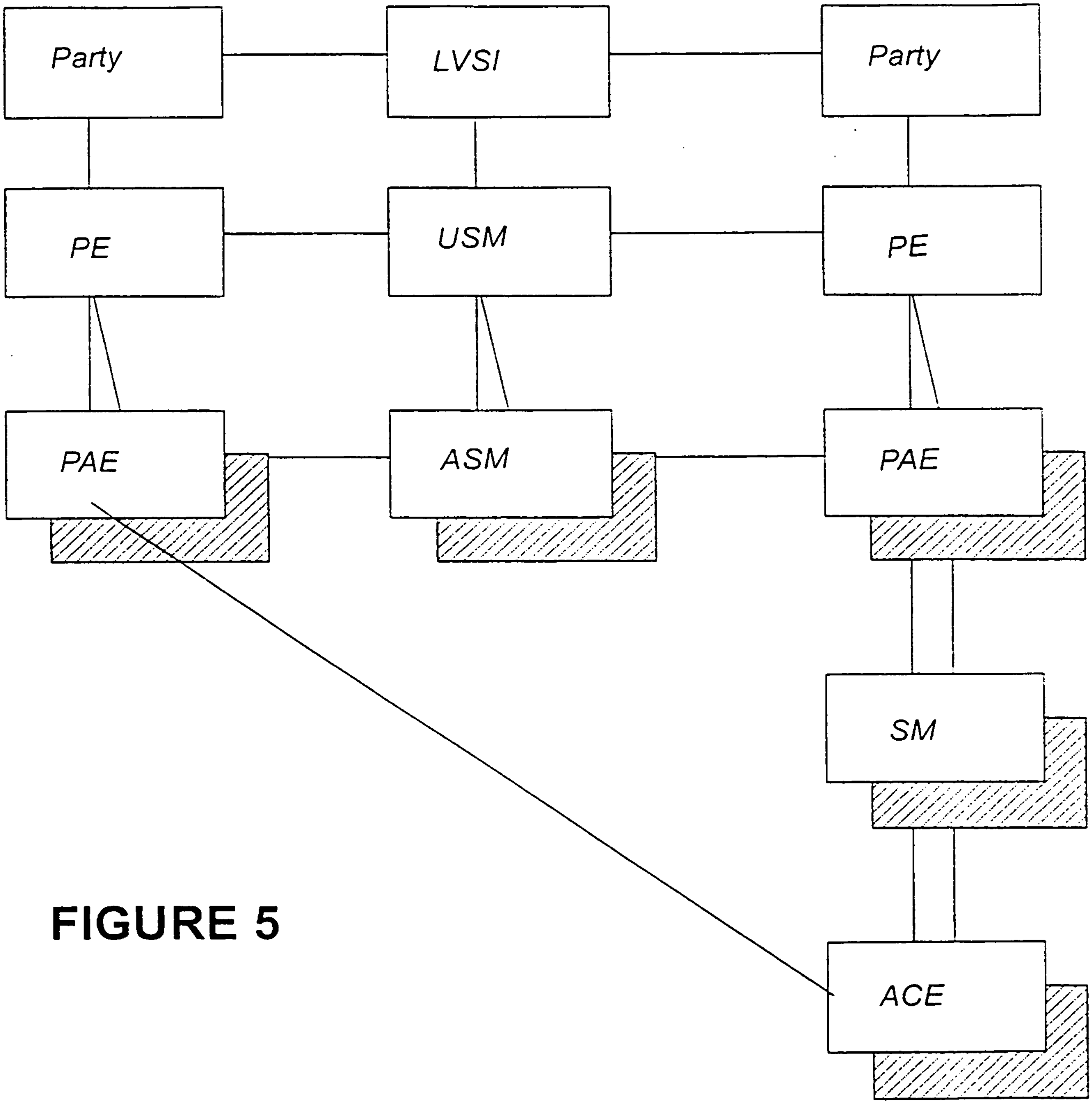


FIGURE 5

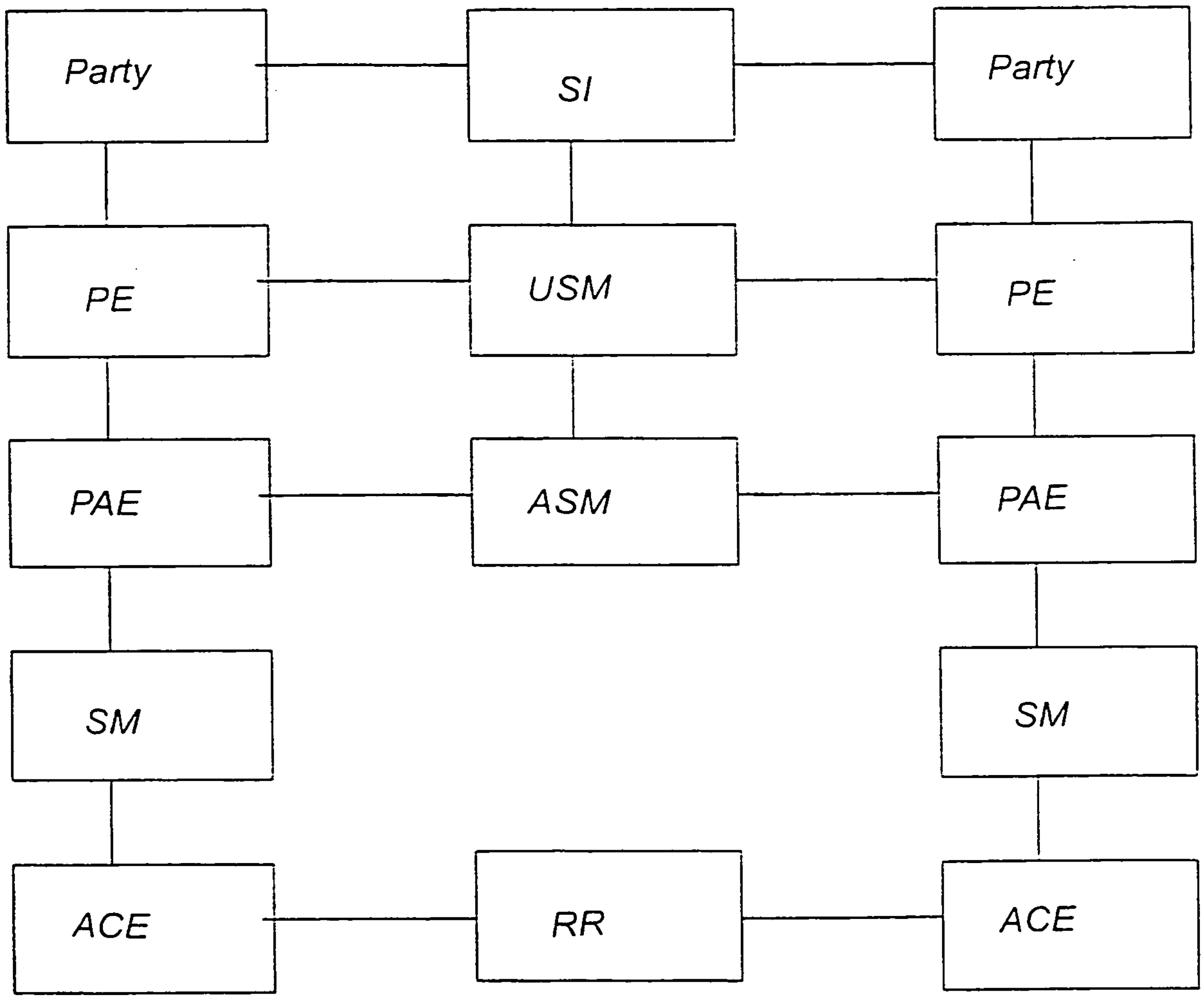


FIGURE 6

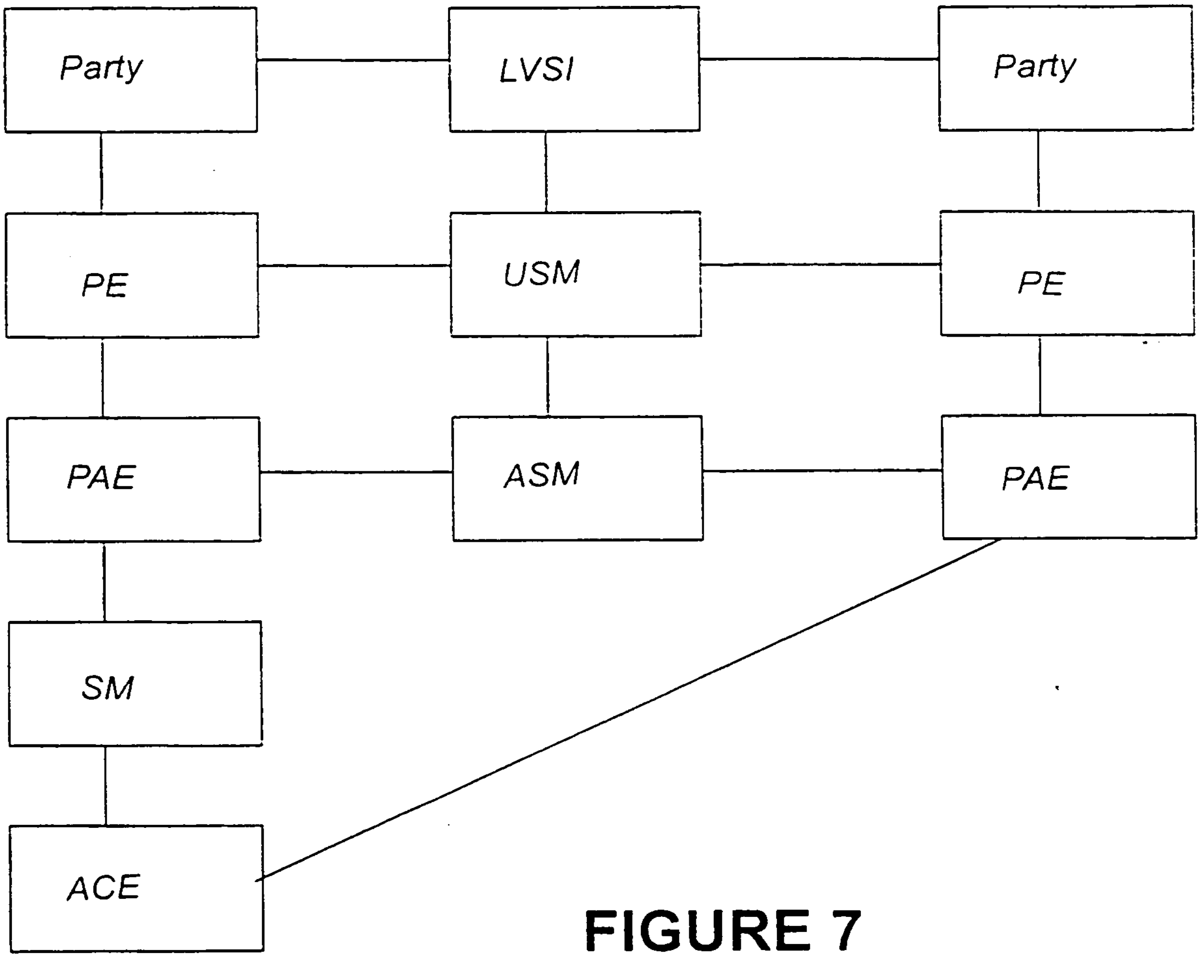


FIGURE 7

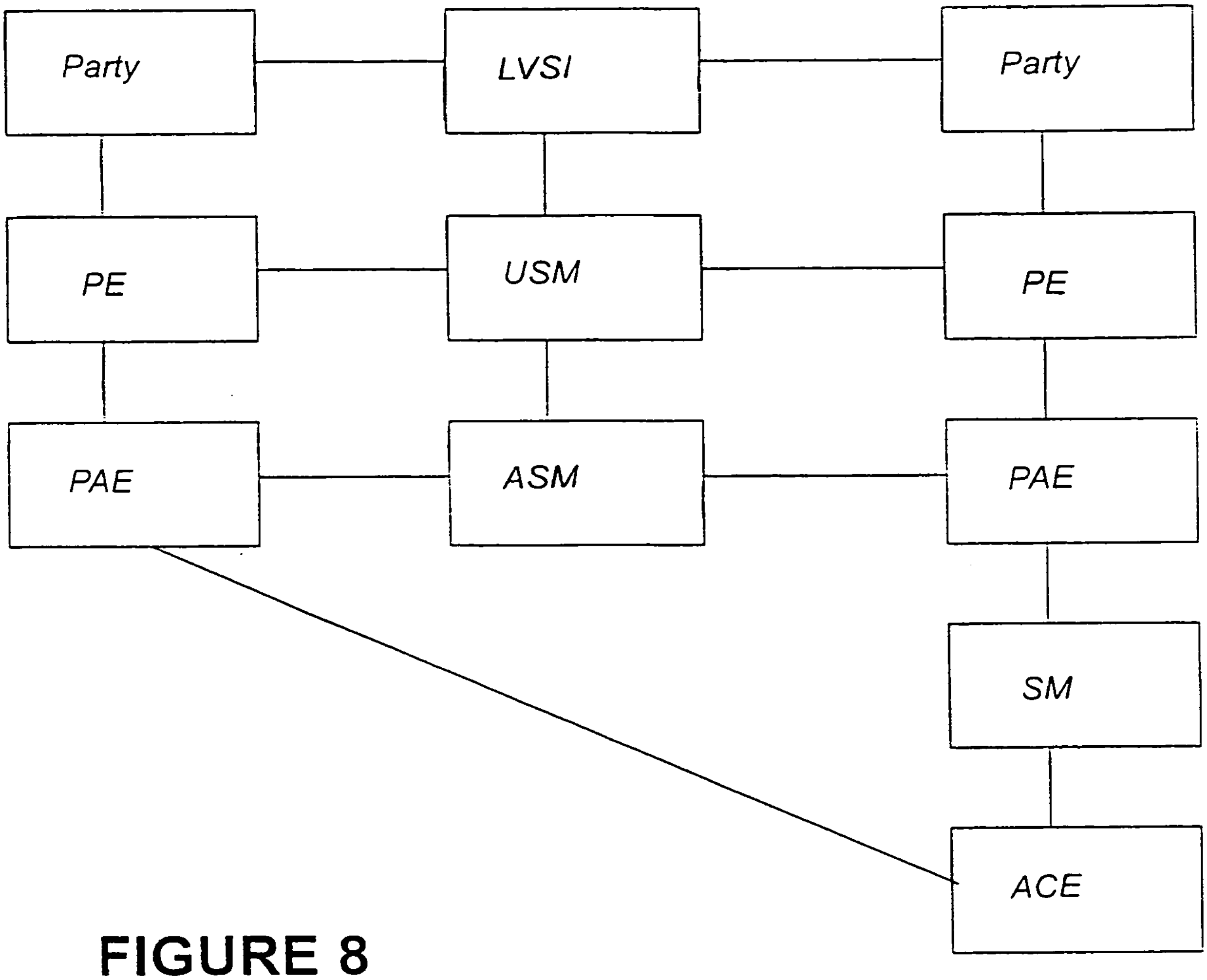
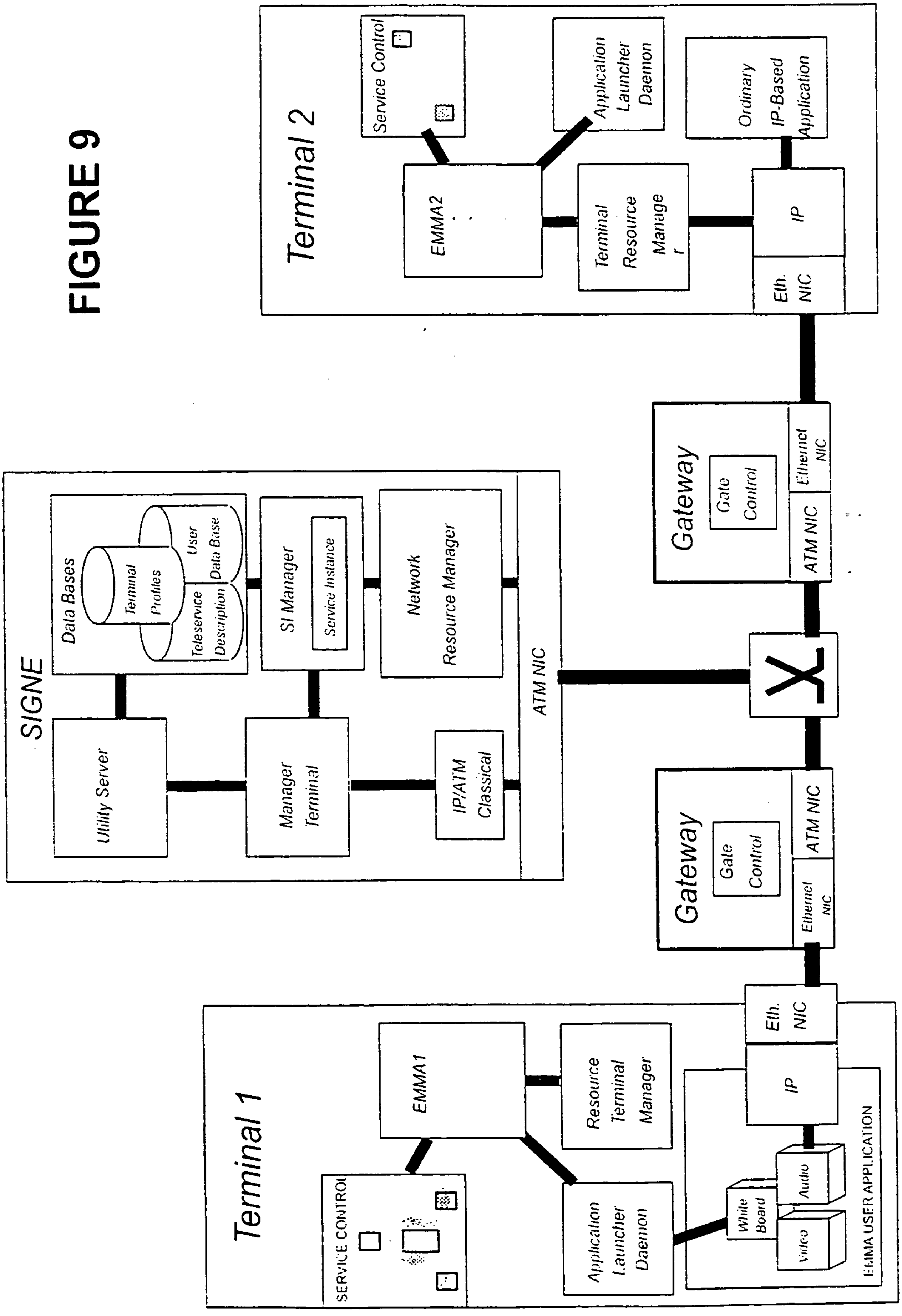


FIGURE 8

FIGURE 9



SUBSTITUTE SHEET (RULE 26)

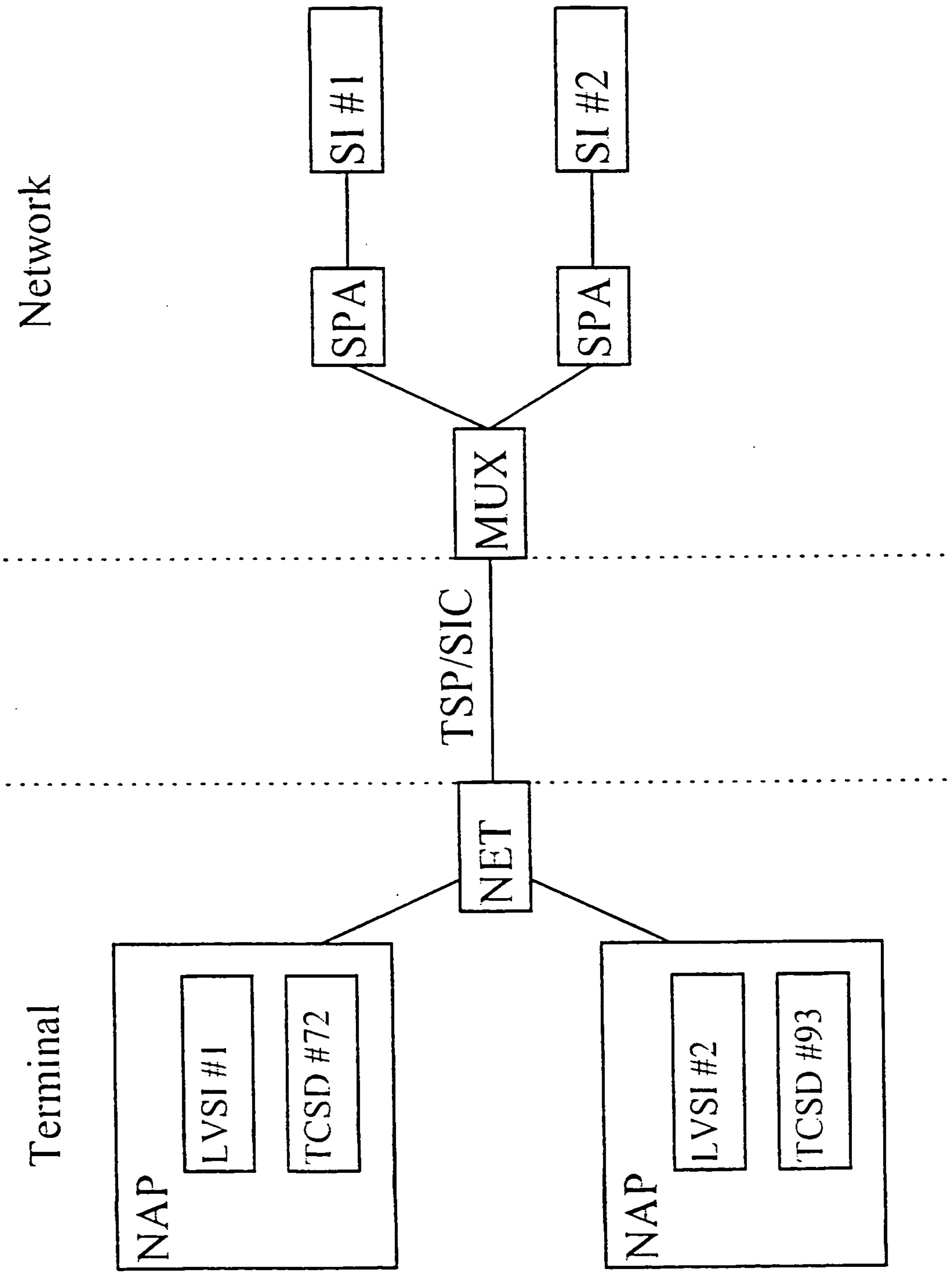

Network
SI #1
SI #2
SPA
SPA
MUX
TSP/SIC
NET
Terminal
NAP
LVSI #1
TCSD #72
NAP
LVSI #2
TCSD #93

FIGURE 10

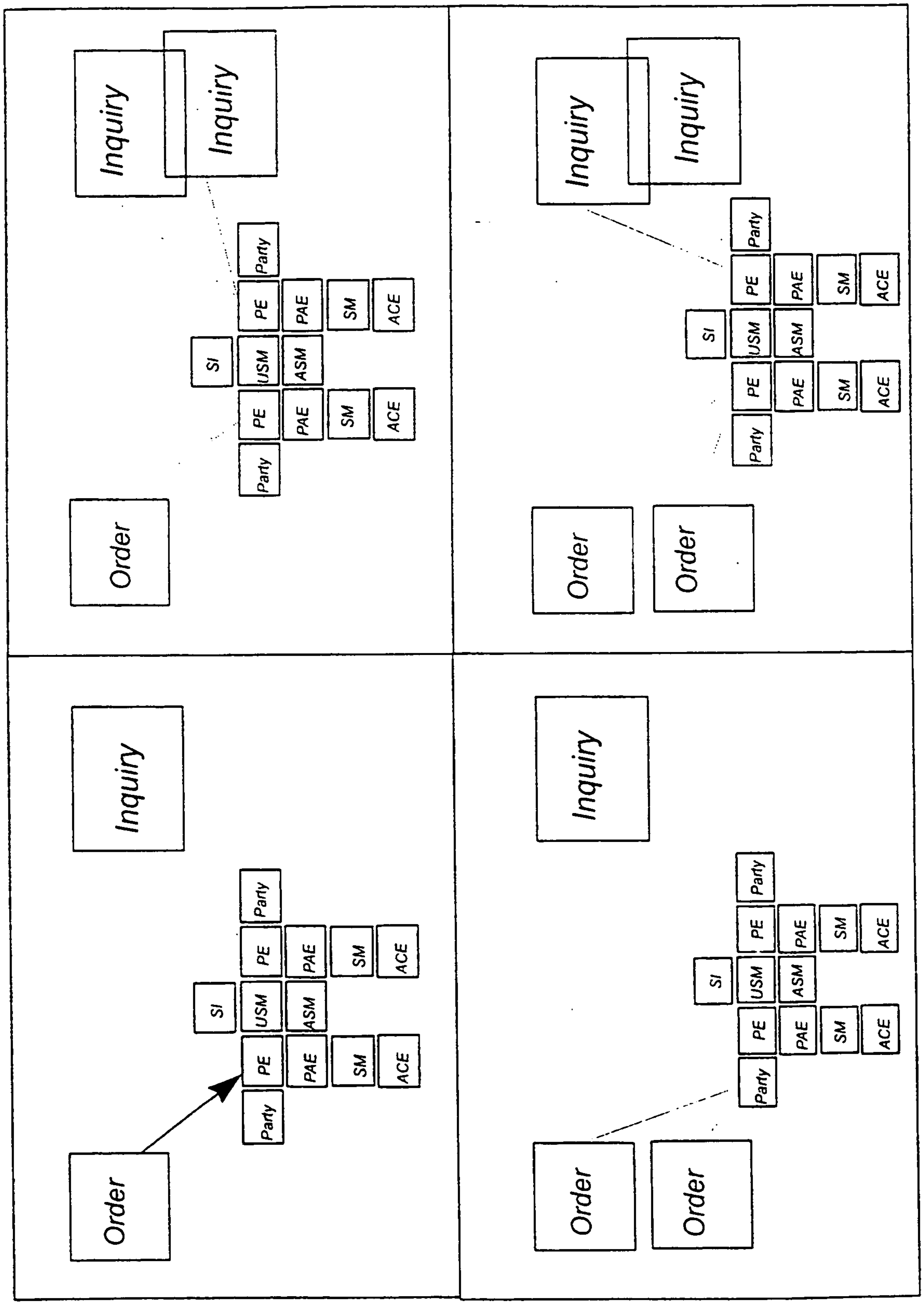

Inquiry
Inquiry
Party
PE
PAE
SM
ACE
SI
USM
ASM
PE
PAE
SM
ACE
Party
Order
Inquiry
Inquiry
Party
PE
PAE
SM
ACE
SI
USM
ASM
PE
PAE
SM
ACE
Party
Order
Order
Inquiry
Party
PE
PAE
SM
ACE
SI
USM
ASM
PE
PAE
SM
ACE
Party
Order
Inquiry
Party
PE
PAE
SM
ACE
SI
USM
ASM
PE
PAE
SM
ACE
Party
Order
Order

FIGURE 11

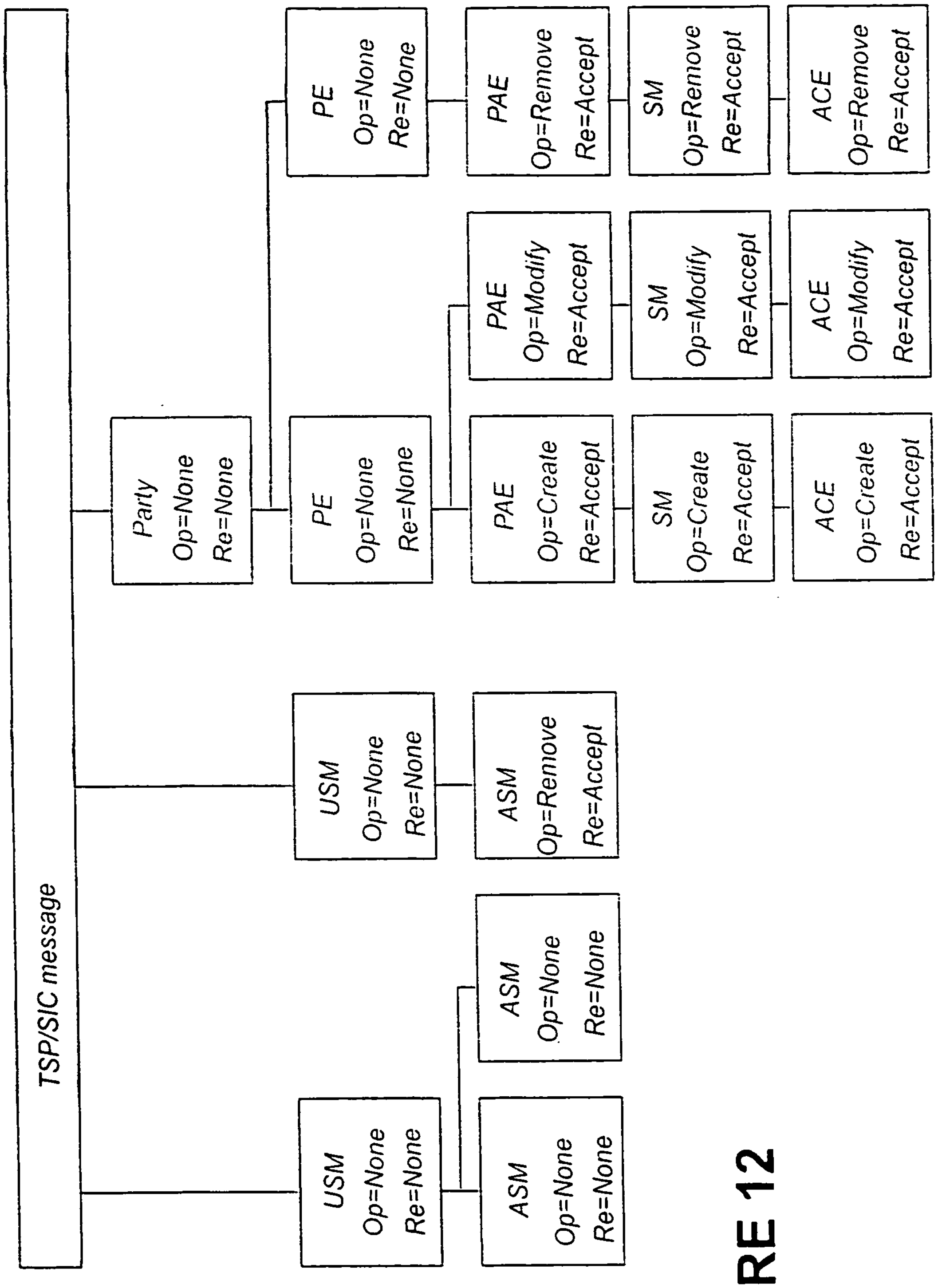

TSP/SIC message
USM
Op=None
Re=None
ASM
Op=None
Re=None
ASM
Op=None
Re=None
USM
Op=None
Re=None
ASM
Op=Remove
Re=Accept
Party
Op=None
Re=None
PE
Op=None
Re=None
PAE
Op=Create
Re=Accept
SM
Op=Create
Re=Accept
ACE
Op=Create
Re=Accept
PAE
Op=Modify
Re=Accept
SM
Op=Modify
Re=Accept
ACE
Op=Modify
Re=Accept
PE
Op=None
Re=None
PAE
Op=Remove
Re=Accept
SM
Op=Remove
Re=Accept
ACE
Op=Remove
Re=Accept

FIGURE 12

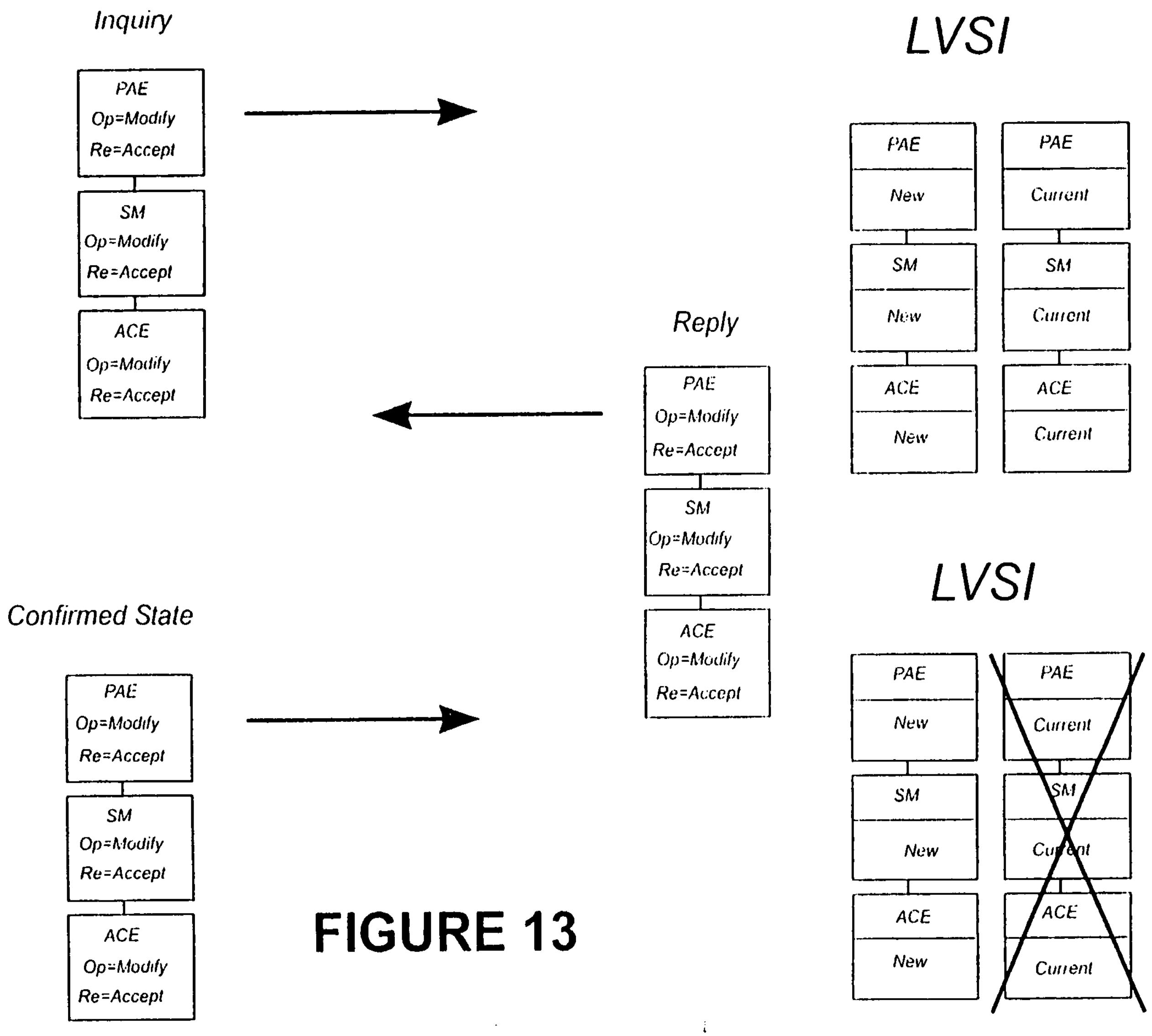

Inquiry
PAE
Op=Modify
Re=Accept
SM
Op=Modify
Re=Accept
ACE
Op=Modify
Re=Accept
Reply
PAE
Op=Modify
Re=Accept
SM
Op=Modify
Re=Accept
ACE
Op=Modify
Re=Accept
LVSI
PAE
New
SM
New
ACE
New
PAE
Current
SM
Current
ACE
Current
Confirmed State
PAE
Op=Modify
Re=Accept
SM
Op=Modify
Re=Accept
ACE
Op=Modify
Re=Accept
LVSI
PAE
New
SM
New
ACE
New
PAE
Current
SM
Current
ACE
Current

FIGURE 13

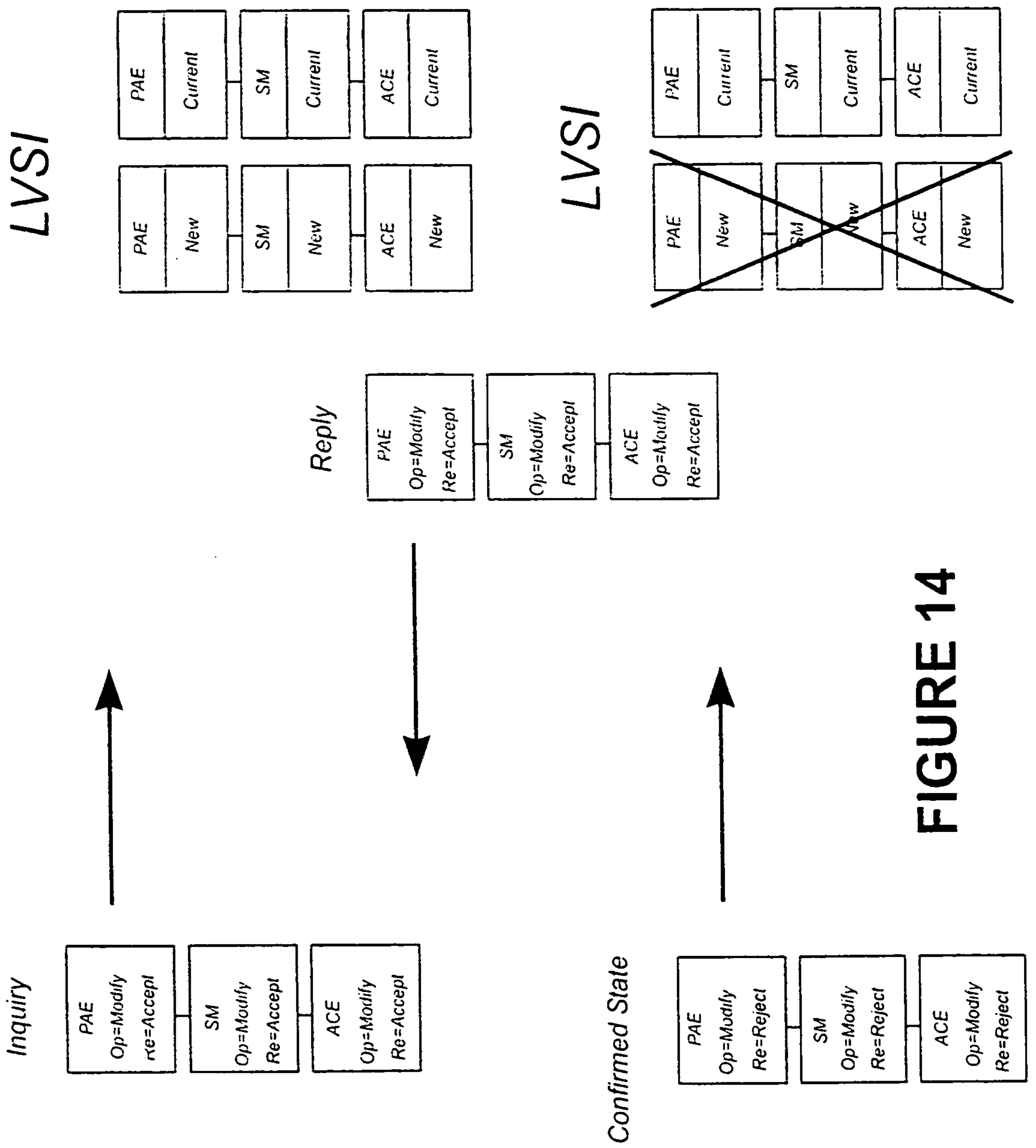

Inquiry
PAE
Op=Modify
Re=Accept
SM
Op=Modify
Re=Accept
ACE
Op=Modify
Re=Accept
LVSI
PAE
New
SM
New
ACE
New
PAE
Current
SM
Current
ACE
Current
Reply
PAE
Op=Modify
Re=Accept
SM
Op=Modify
Re=Accept
ACE
Op=Modify
Re=Accept
Confirmed State
PAE
Op=Modify
Re=Reject
SM
Op=Modify
Re=Reject
ACE
Op=Modify
Re=Reject
LVSI
PAE
New
SM
New
ACE
New
PAE
Current
SM
Current
ACE
Current

FIGURE 14

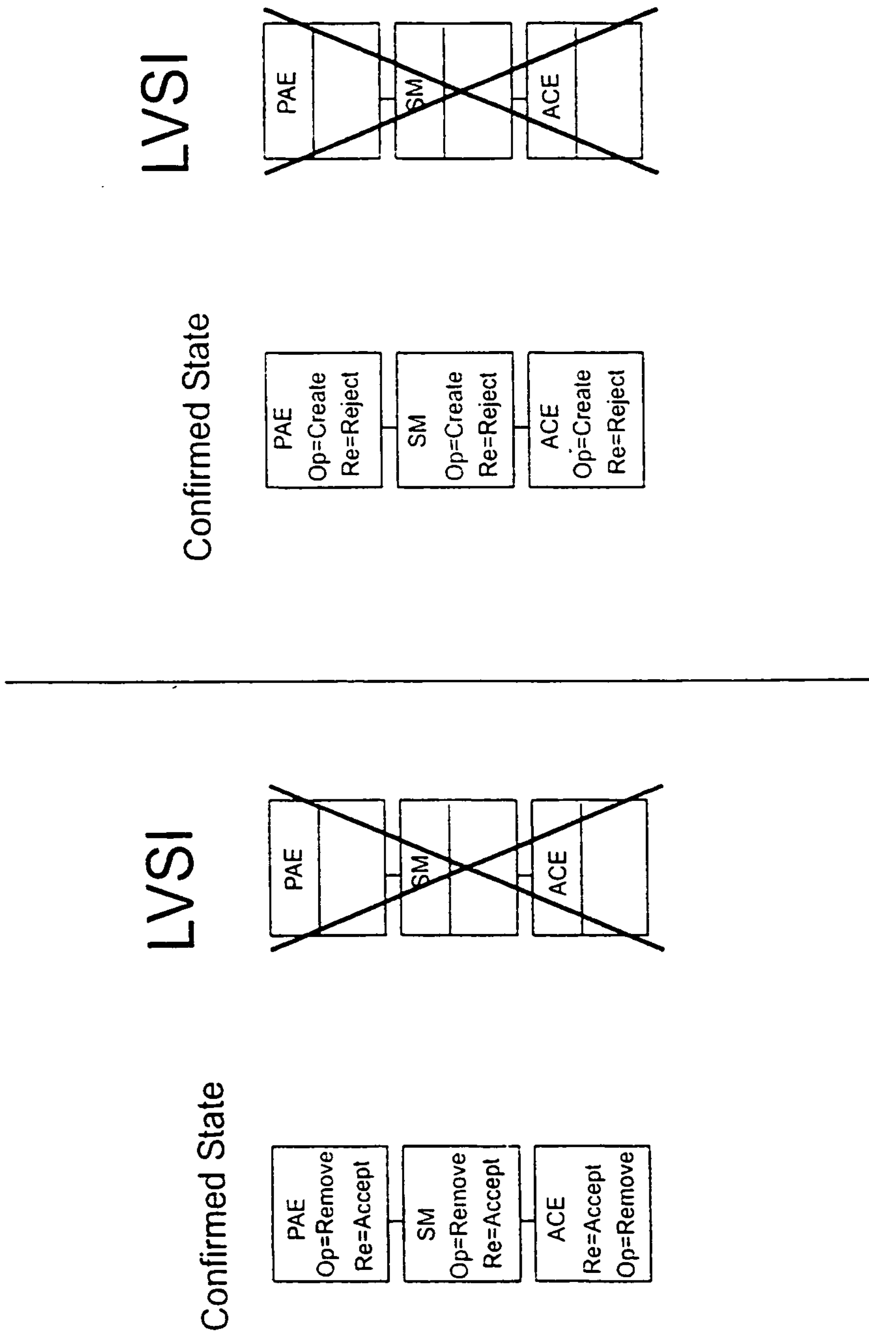

Confirmed State
PAE
Op=Remove
Re=Accept
SM
Op=Remove
Re=Accept
ACE
Re=Accept
Op=Remove
LVSI
PAE
SM
ACE
Confirmed State
PAE
Op=Create
Re=Reject
SM
Op=Create
Re=Reject
ACE
Op=Create
Re=Reject
LVSI
PAE
SM
ACE

FIGURE 15

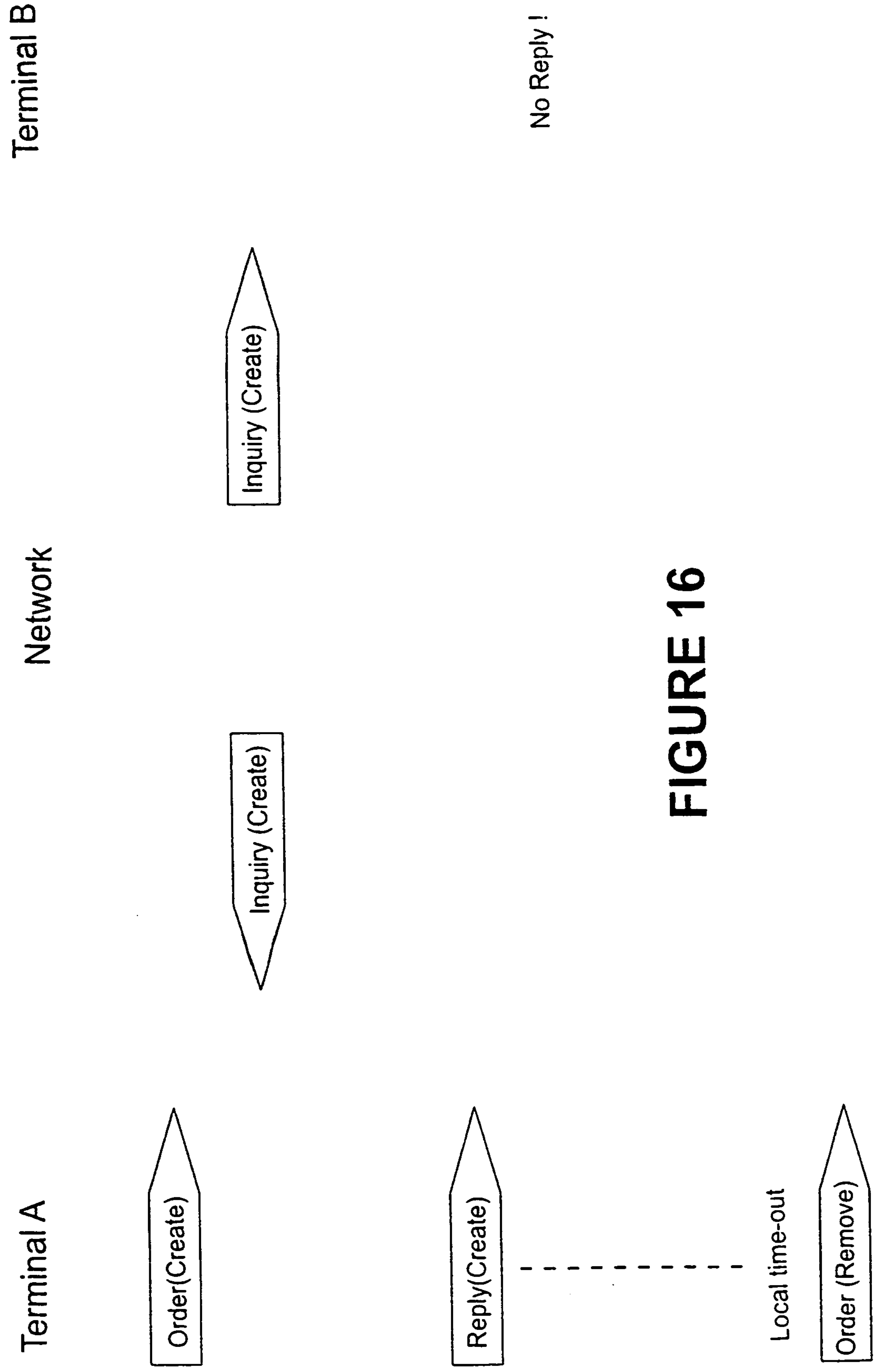

Terminal A
Network
Terminal B
Order(Create)
Inquiry (Create)
Inquiry (Create)
No Reply !
Reply(Create)
Local time-out
Order (Remove)

FIGURE 16

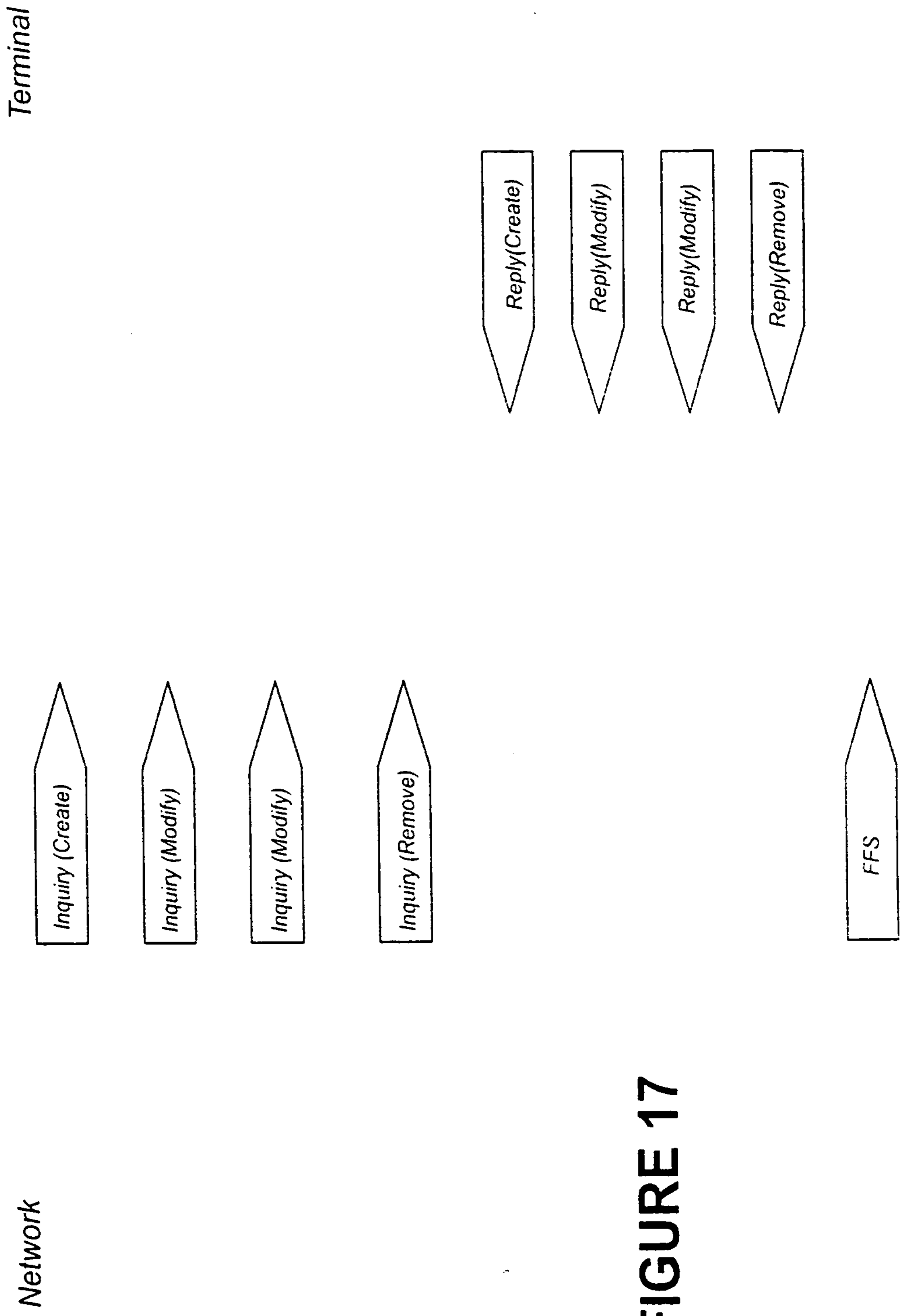

Terminal
Network
Inquiry (Create)
Inquiry (Modify)
Inquiry (Modify)
Inquiry (Remove)
Reply(Create)
Reply(Modify)
Reply(Modify)
Reply(Remove)
FFS

FIGURE 17

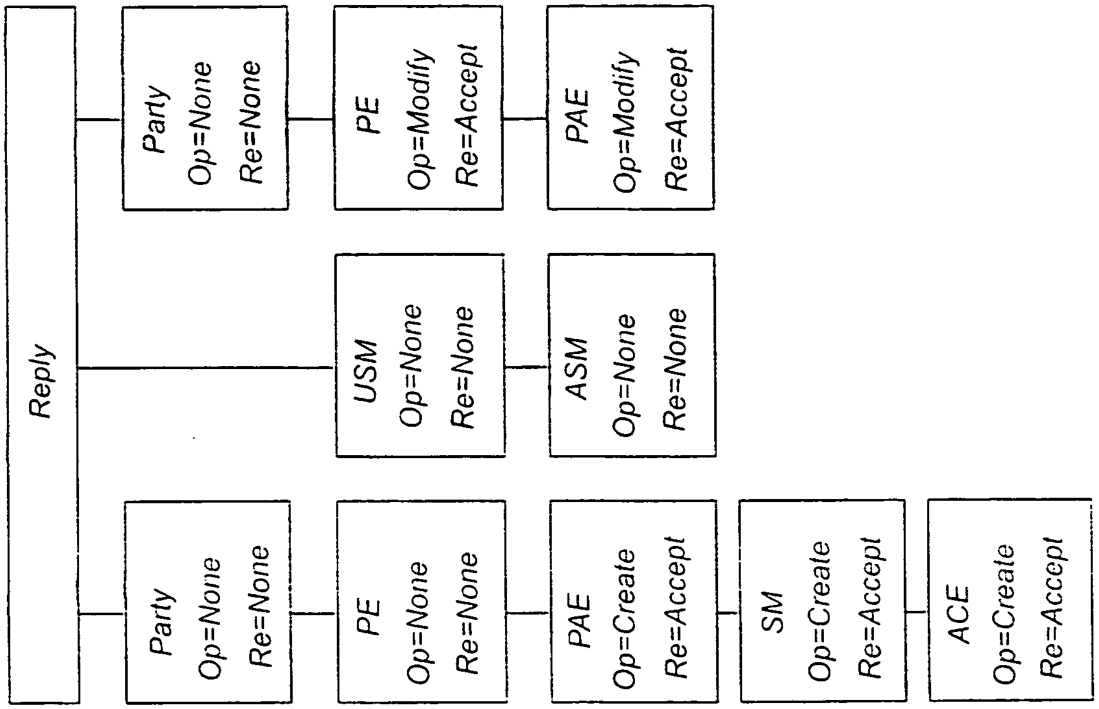
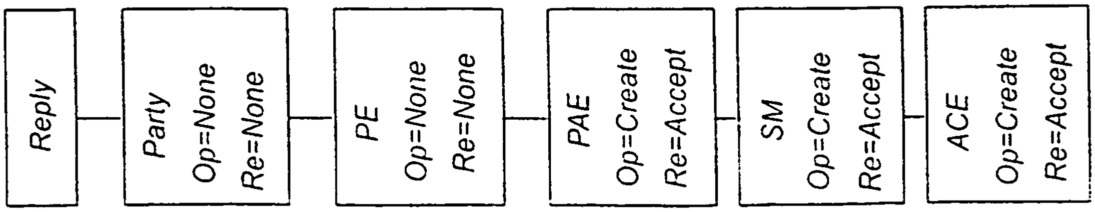

Reply
Party
Op=None
Re=None
PE
Op=Modify
Re=Accept
PAE
Op=Modify
Re=Accept
USM
Op=None
Re=None
ASM
Op=None
Re=None
Party
Op=None
Re=None
PE
Op=None
Re=None
PAE
Op=Create
Re=Accept
SM
Op=Create
Re=Accept
ACE
Op=Create
Re=Accept

FIGURE 18


Reply
Party
Op=None
Re=None
PE
Op=None
Re=None
PAE
Op=Create
Re=Accept
SM
Op=Create
Re=Accept
ACE
Op=Create
Re=Accept

Calling Terminal		Network		Called Terminal
Calling sends a Order				
--- Order ---->				
		Network processes the Order and sends Inquiries		
		<--- Inquiry --->		
The terminal allocates the ABB and sends a Reply				
--- Reply --->				The terminal ask the user to join. If he joins. the terminal allocates the ABB and sends a Reply
		Network updates the SI		
				<--- Reply ---
		Network updates the SI and finds that the minimum demands for setting up a SI is fullfilled , tells the resources in the network to activate. The two terminals are now connected. Network sends Confirmed State to both terminals.		
		<--- CS --->		
The terminal activates the ABB and starts to communicate with the other terminal				The terminal activates the ABB and starts to communicate with the other terminal

FIGURE 19

Calling Terminal		Network		Called Terminal
The calling terminal is communicating with the other terminal and deside to terminate the SI. An Order about termination is sent to the network.				The called terminal is communicating with the other terminal.
--- Order ---->				
		Network processes the Order and sends Inquiries to the involved partys.		
		<--- Inquiry --->		
The terminal tells the ABB to finish and when it is done it sends a Reply.				
--- Reply --->				The terminal knows it is time to terminate the SI, finish the current transfer and exiting the ABB and sends a reply to the network.
		Network updates the SI		
				<--- Reply ---
		Network updates the SI, deactivates the resources, disconnects the terminals and sends Confirmed State.		
		<--- CS --->		
The Confirmed State message tells the terminal that the SI has ended and the LVSI can be thrown away.				The Confirmed State message tells the terminal that the SI has ended and the LVSI can be thrown away.

FIGURE 20

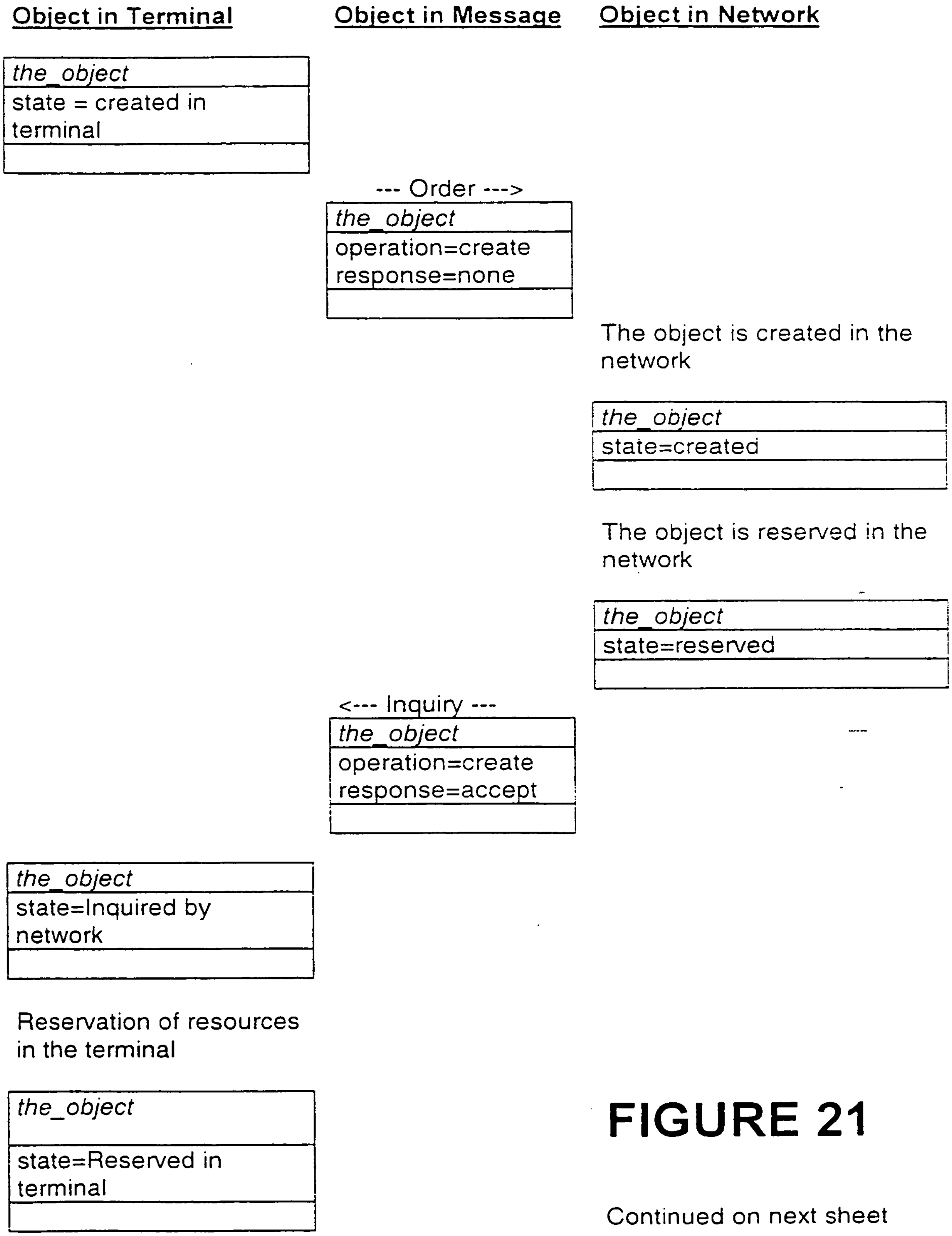


FIGURE 21

Continued on next sheet

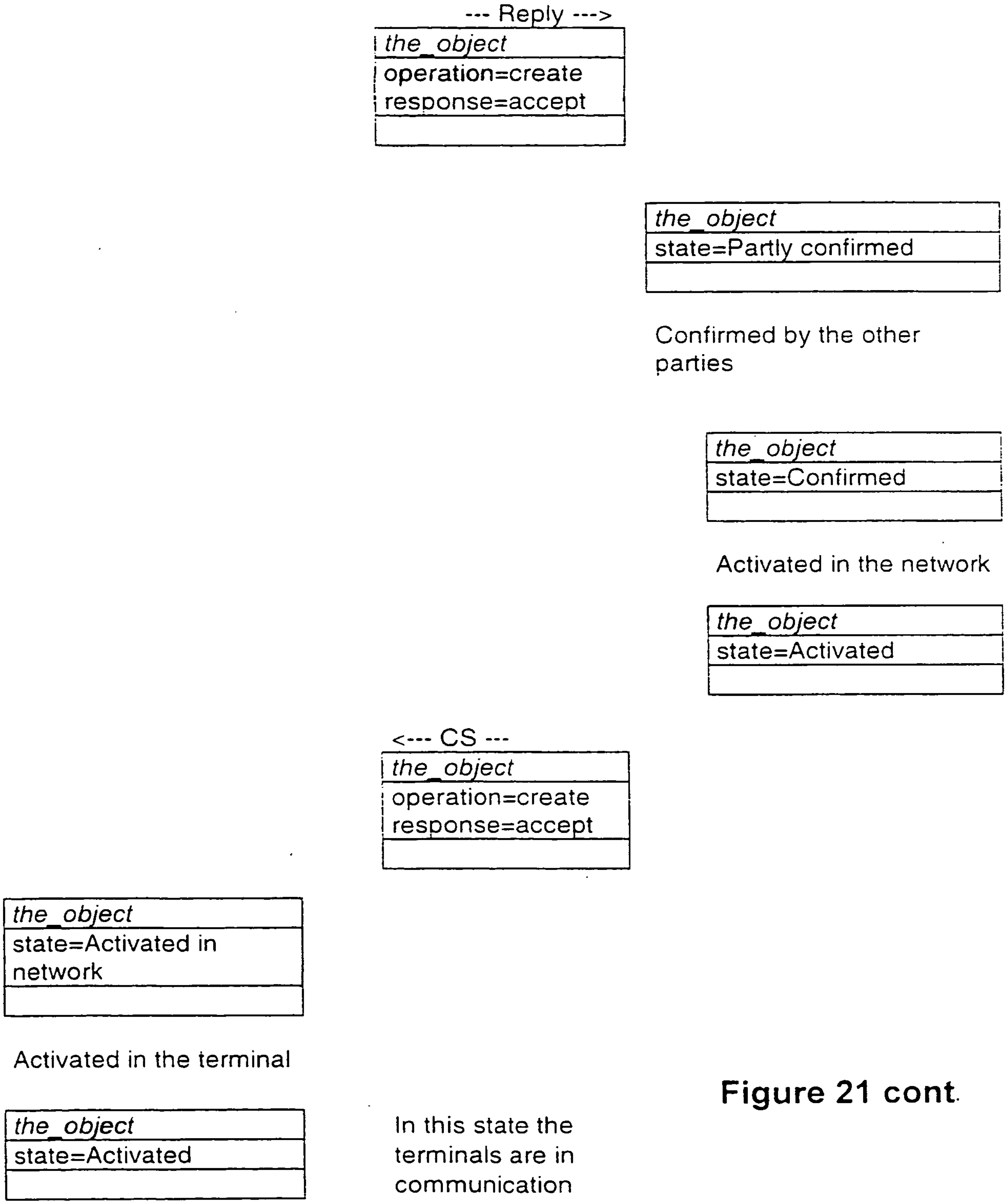
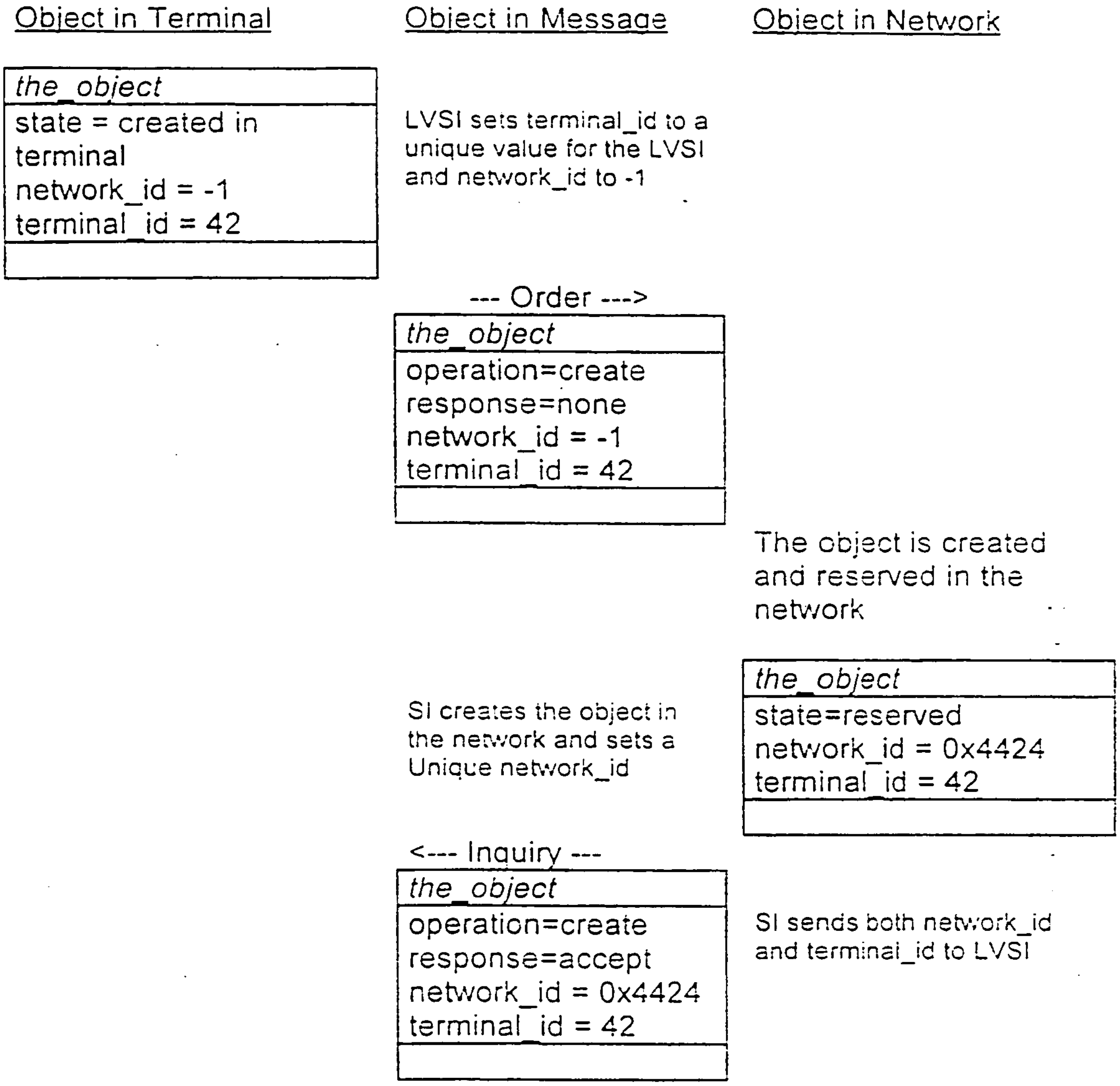


Figure 21 cont.



Object in Terminal

Object in Message

Object in Network

the_object

state = created in terminal
network_id = -1
terminal_id = 42

LVSI sets terminal_id to a unique value for the LVSI and network_id to -1

--- Order --->

the_object

operation=create
response=none
network_id = -1
terminal_id = 42

The object is created and reserved in the network

SI creates the object in the network and sets a Unique network_id

the_object

state=reserved
network_id = 0x4424
terminal_id = 42

<--- Inquiry ---

the_object

operation=create
response=accept
network_id = 0x4424
terminal_id = 42

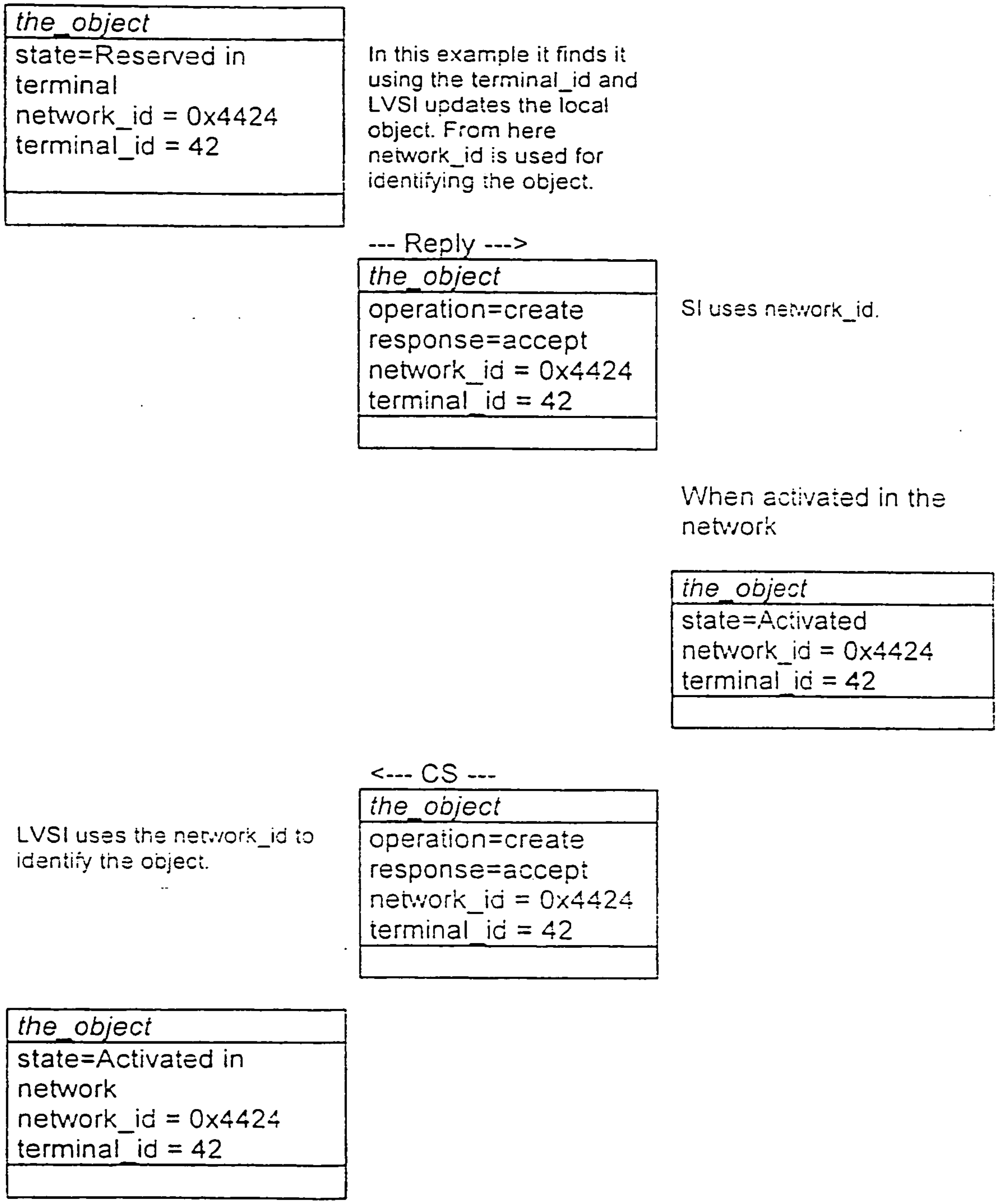
SI sends both network_id and terminal_id to LVSI

Reservation in the terminal

LVSI looks for an object with the network_id, if LVSI can't find any object it looks for an object with the terminal_id, if LVSI can't find the object it creates the object. In this example it finds it using the terminal_id

FIGURE 22

Continued on next sheet



the_object
state=Reserved in terminal
network_id = 0x4424
terminal_id = 42

In this example it finds it using the terminal_id and LVSI updates the local object. From here network_id is used for identifying the object.

--- Reply --->

the_object
operation=create
response=accept
network_id = 0x4424
terminal_id = 42

SI uses network_id.

When activated in the network

the_object
state=Activated
network_id = 0x4424
terminal_id = 42

<--- CS ---

the_object
operation=create
response=accept
network_id = 0x4424
terminal_id = 42

LVSI uses the network_id to identify the object.

the_object
state=Activated in network
network_id = 0x4424
terminal_id = 42

Activated in the terminal

FIGURE 22 cont.

Table 1
Information Transferred through Various TSMS Interfaces

Interface	Information Transferred
between EMMA/SIGNE	a part of the teleservice object model
between EMMA/SC	a part of the teleservice object model
between EMMA/APLD	name and parameters of the applications(s) involved in the teleservice
between EMMA/TRM	specification of terminal resources required by the teleservice
between SIGNE/NRM	specification of network resources required by the teleservice

Table 2
Typical Messages for Establishing a Teleservice Session between Two Parties

	Message Name	Source	Destination
1	SC-Order	SC1	EMMA1
2	UA-Check	EMMA1	APLD1
3	UA-Check-Response	APLD1	EMMA1
4	Terminal-Check	EMMA1	TRM1
5	Terminal-Check-Resp.	TRM1	EMMA1
6	Order	EMMA1	SIGNE
7	Network-Check	SIGNE	NRM
8	Network -Check-Resp.	NRM	SIGNE
9	Inquiry	SIGNE	EMMA1. EMMA2
10a	UA-Check	EMMA2	APLD2
11a	UA-Check-Response	APLD2	EMMA2
12a	Terminal-Check	EMMA2	TRM2
13a	Terminal-Check-Resp.	TRM2	EMMA2
14a	SC-Inquiry	EMMA2	SC2
15a	SC-Reply	SC2	EMMA2
16a	Reply	EMMA2	SIGNE
10b	SC-Inquiry	EMMA1	SC1
11b	SC-Reply	SC1	EMMA1
12b	Reply	EMMA1	SIGNE
17	Network-Start	SIGNE	NRM
18	Network -Start-Resp.	NRM	SIGNE
19	Confirmed State	SIGNE	EMMA1, EMMA2
20a	UA-Start	EMMA1	APLD1
21a	UA-Start-Response	APLD1	EMMA1
22a	Terminal-Start	EMMA1	TRM1
23a	Terminal-Start-Resp.	TRM1	EMMA1
24a	SC-CS	EMMA1	SC1
20b	UA-Start	EMMA2	APLD2
21b	UA-Start-Response	APLD2	EMMA2
22b	Terminal-Start	EMMA2	TRM2
23b	Terminal-Start-Resp.	TRM2	EMMA2
24b	SC-CS	EMMA2	SC2

Table 3: Structure of a TSP Message

Syncword=HELLO SIGNE! (12 bytes)	prot = TSP/SIC (4 bytes)	tsp_rev (4 bytes)	org_id (8 bytes)	dest_id (8 bytes)
username (20 bytes)	Hostname (60 bytes)	tot_size_of_msg (4 bytes)	prot_data = SIC_MSG (xx bytes)	

Table 4: Attributes in Class TSP_MSG

No	Type	Name	Description
1	const char[12]	Syncword	Value: "HELLO SIGNE!"
2	int	prot	Protocol descriptor. See table.
3	const int	tsp_rev	Revision of the TSP_MSG. This rev = 3
4	int [2]	org_id	Originating Instance identifyer and return adress. Org_id=0 is not valid.
5	int [2]	dest_id	Destination identifyer. if 0 create a new instance.
6	char [20]	username	username in the terminal (Never the username of the process owner in the network (SIGNE))
7	char [60]	hostname	hostname of the terminal. (OBS! Max size in Solaris 2.5 is 256 bytes)
8	int	tot_size_of_msg	Size of the TSP message in bytes
9	char [xx]	prot_data	PDU. size is (tot_size_of_msg - "other attributs in classTSP_MSG").

Table 5: Protocol Descriptor for TSP Message

Value	Name	Description
0	Undefined	Undefined
1	TSP/SIC	Service Instance control protocol
2	TSP/TCSD	TSC description protocol
3	TSP/DIR	Directory protocol
4	TSP/Term Profile	Terminal profile protocol
5	TSP/REG	Registration protocol
6	TSP/MUX	For control messages between the muxes.
7	TSP/ITCSD	Signe Internal TSC description protocol
8 to max int	-	ffs

Table 6: PDU for TSP/SIC

message type (4 bytes)	sic_rev (4 bytes)	si_head (xx bytes)	usm_leg (xx bytes)	party_leg (xx bytes)

Table 7: Attributes in Class SIC_MSG

No	Type	Name	Description
1	int	message type	See Table
2	const int	sic_rev	Revision of TSP/SIC, This rev = 4
3	class SI_HEAD	si_head	See class SI_HEAD
4	char [xx]	usm_leg	PDU for USM leg
5	char [xx]	party_leg	PDU for PARTY leg

Table 8: Table for Message type

Value	Name	Description
0	Undefined	Undefined
1	ORDER	Sent from Terminal
2	INQUIRY	Sent from Network
3	REPLY	Sent from Terminal
4	CS	Sent from Network
5 to max int	-	ffs

Table 9: PDU for USM Leg

size_of_usm_leg (4 byte)	elements ((size_of_usm_leg - 4) bytes)

Table 10: Attributes in Class USM_LEG

No	Type	Name	Description
1	int	size_of_usm_leg	Size of the USM leg in bytes
2	char [xx]	elements	PDU for USMs and ASMs

Table 11: PDU for PARTY_LEG

size_of_party_leg (4 byte)	elements ((size_of_party_leg - 4) bytes)

Table 12: Attributes in Class PART_LEG

No	Type	Name	Description
1	int	size_of_party_leg	Size of the Party leg in bytes
2	char [xx]	elements	PDU for PARTY. PEs. PAEs. SMs and ACEs

Table 13: Attributes in Class ELEMENT_HEAD_IN_LEG

No	Type	Name	Description
1	int	element_type	Type of Element. See table.
2	int	network_id	Unique for the Service Instance. Primary id.
3	int	terminal_id	Unique for LVSI of initiator. Used of LVSI until network_id not equal to -1.
4	int	operation	Operation on the object. See text and table.
5	int	response	Response to the operation. See text and table.
6	int	cause	Cause value. See table.

Table 14: Table for Element Type

Value	Name	Description
0	Undefined	Undefined
1	USM_OBJECT	
2	ASM_OBJECT	
3	PARTY_OBJECT	Not used in TSP/TCSD
4	PE_OBJECT	
5	PAE_OBJECT	
6	SM_OBJECT	
7	ACE_OBJECT	
8	SI_HEAD_OBJECT	Not used in TSP/TCSD
9	TCSD_HEAD_OBJECT	Not used in TSP/SIC
10	PARTY_TYPE_OBJECT	Not used in TSP/SIC
11 to max int	-	ffs

Table 15: Table for Operation

Value	Name	Description
0	Undefined	Undefined
1	NONE	See text.
2	CREATE	See text.
3	MODIFY	See text.
4	REMOVE	See text.
5 to max int	-	ffs

Table 16: Table for Response

Value	Name	Description
0	Undefined	Undefined
1	NONE	See text.
2	ACCEPT	See text.
3	REJECT	See text.
4 to max int	-	ffs

Table 18: PDU for SI_HEAD

head (28 bytes)	Service Utility ref (4 bytes)	TCS_ref (4 byte)	TCS_version (4 bytes)

Table 19: Attributes in Class SI_HEAD

No	Type	Name	Description
1	class ELEMENT_HEAD_IN_LEG	head	Common attributs
2	int	Service_Utility_ref	End to end, reference for the utility of the service
3	int	TCS_ref	Reference to the TCSD
4	int	TCS_version	Version of the TCSD

Table 20: PDU for USM_ELEMENT

head (28 bytes)	type (4 bytes)	presence (4 bytes)

Table 21: Attributes in class USM_ELEMENT

No	Type	Name	Description
1	class ELEMENT_HEAD_IN_LEG	head	Common attributs
2	int	type	See table for USM::Type in TSPTCSD.DOC
3	int	presence	See table

Table 22: Table for presence in USM

Value	Name	Description
0	Undefined	Undefined
1	Mandatory	
2	Optional	
3 to max int	-	ffs

Table 23: PDU for ASM_ELEMENT			
head (28 bytes)	type (4 bytes)	presence (4 bytes)	quality (4 bytes)

Table 24: Attributes in class ASM_ELEMENT			
No	Type	Name	Description
1	class ELEMENT_HEAD_IN_LEG	head	Common attributs
2	int	type	See table for ASM::Type in TSPTCSD.DOC
3	int	presence	See table
4	int	quality	ffs.

Table 25: Table for presence in ASM		
Value	Name	Description
0	Undefined	Undefined
1	Mandatory	
2	Optional	
3 to max int	-	ffs

Table 26: PDU for PARTY_ELEMEN			
head (28 bytes)	type (4 bytes)	presence (4 bytes)	party_id (100 bytes)

Table 27: Attributes in class PARTY_ELEMENT			
No	Type	Name	Description
1	class ELEMENT_HEAD_IN_LEG	head	Common attributs
2	int	type	Reference to a Party_type in TCSD. See table for PARTY_TYPE::Role in TSPTCSD.DOC
3	int	presence	See table
4	class PARTY_ID	party_id	ffs.

Table 28: Table for presence in PARTY_ELEME

Value	Name	Description
0	Undefined	Undefined
1	Mandatory	
2	Optional	
3 to max int	-	ffs

Table 29: PDU for PARTY_ID

name (20 bytes)	terminalname (60 bytes)	E164address (20 bytes)		

Table 30: Attributes in class PARTY_ID

No	Type	Name	Description
1	char[20]	name	Name. can be same value as username in the terminal.
2	char[60]	terminalname	Terminalname, can be same value as the hostname of the terminal.
3	unsigned char[20]	E164address	To be used when a terminal don't support SIGNE and TSP, like a Q.2931 terminal.

Table 31: PDU for PE_ELEMENT

head (28 bytes)	usm_ref (4 bytes)	usm_terminal_ref (4 bytes)	presence (4 bytes)

Table 32: Attributes in class PE_ELEMENT

No	Type	Name	Description
1	class ELEMENT_HEAD_IN_LEG	head	Common attributs
2	int	usm_ref	Reference to USM
3	int	usm_terminal_ref	Ref to USM using terminal_id
4	int	presence	See table

Table: 33 Table for presence in PE_ELEMENT		
Value	Name	Description
0	Undefined	Undefined
1	Mandatory	
2	Optional	
3 to max int	-	ffs

Table 34: PDU for PAE_ELEMENT				
head (28 bytes)	asm_ref (4 bytes)	asm_terminal_ref (4 bytes)	presence (4 bytes)	direction (4 bytes)

Table 35: Attributes in class PAE_ELEMENT			
No	Type	Name	Description
1	class ELEMENT_HEAD_IN_LEG	head	Common attributs
2	int	asm_ref	Reference to ASM
3	int	asm_terminal_ref	Ref to ASM usning terminal_id
4	int	presence	See table
5	int	direction	Mapping direction for the PAE

Table 36: Table for presence in PAE_ELEMEN		
Value	Name	Description
0	Undefined	Undefined
1	Mandatory	
2	Optional	
3 to max int	-	ffs

Table 37: Table for direction in PAE_ELEMENT		
Value	Name	Description
0	Undefined	Undefined
1	Send	
2	Receive	
3	Send_and_Receive	
4	None	
5 to max int	-	Undefined

Table 38: PDU for SM_ELEMENT

head (28 bytes)	type (4 bytes)	ABBparam (256 bytes)			

Table 43: Attributes in class TRAFFIC_DESCRIPTOR

No	Type	Name	Description
1	int	Type	Traffic Class Type. See table belowe.
2	int	Param	Parameter for the Traffic Class.

Table 44: Table for Type in class TRAFFIC_DESCRIPTOR

Value	Name	Description
0	Unknown	defined for unknown
1	Free	Telia Free Class.
2	Guaranteed	Telia Guaranteed Class.
3 to max int	-	ffs

Table 45: The meaning of the attributes Operation, Response and Cause in an Order

Operation	Response	Cause	Valid	Description
None	None	None	Valid	Object exist but no action is needed.
None	None	Value	Not Valid	
None	Accept	None	Not Valid	
None	Accept	Value	Not Valid	
None	Reject	None	Not Valid	
None	Reject	Value	Not Valid	
Create	None	None	Valid	In Order to Create a object.
Create	None	Value	Not Valid	
Create	Accept	None	Not Valid	
Create	Accept	Value	Not Valid	
Create	Reject	None	Not Valid	
Create	Reject	Value	Not Valid	
Modify	None	None	Valid	In Order to Modify the object.
Modify	None	Value	Not Valid	
Modify	Accept	None	Not Valid	
Modify	Accept	Value	Not Valid	
Modify	Reject	None	Not Valid	
Modify	Reject	Value	Not Valid	
Remove	None	None	Valid	In Order to Remove the object.
Remove	None	Value	Not Valid	
Remove	Accept	None	Not Valid	
Remove	Accept	Value	Not Valid	
Remove	Reject	None	Not Valid	
Remove	Reject	Value	Not Valid	

Table 46: The meaning of the attributes Operation, Response and Cause in an Inquiry

Operation	Response	Cause	Valid	Description
None	None	None	Valid	Object exist but no action is needed.
None	None	Value	Not Valid	
None	Accept	None	Not Valid	
None	Accept	Value	Not Valid	
None	Reject	None	Not Valid	
None	Reject	Value	Not Valid	
Create	None	None	Valid	Object exist but no action is needed.
Create	None	Value	Not Valid	
Create	Accept	None	Valid	The network accept creation of this object and asks the terminal if it accepts a creation of this object.
Create	Accept	Value	Not Valid	
Create	Reject	None	Not Valid	
Create	Reject	Value	Valid	The network reject creation of this object. This is valid when it is sent to the sender of the Order
Modify	None	None	Valid	Object exist but no action is needed.
Modify	None	Value	Not Valid	
Modify	Accept	None	Valid	The network accept modification of this object and asks the terminal if it accepts a modification of this object.
Modify	Accept	Value	Not Valid	
Modify	Reject	None	Not Valid	
Modify	Reject	Value	Valid	The network reject modification of this object. This is valid when it is sent to the sender of the Order
Remove	None	None	Valid	Object exist but no action is needed.
Remove	None	Value	Not Valid	
Remove	Accept	None	Valid	The network accept removal of this object and asks the terminal if it accepts a removal of this object.
Remove	Accept	Value	Not Valid	
Remove	Reject	None	Not Valid	
Remove	Reject	Value	Valid	The network reject removal of this object. This is valid when it is sent to the sender of the Order

Table 47: The meaning of the attributes Operation, Response and Cause in a Reply

Operation	Response	Cause	Valid	Description
None	None	None	Valid	Object exist but no action is needed.
None	None	Value	Not Valid	
None	Accept	None	Not Valid	
None	Accept	Value	Not Valid	
None	Reject	None	Not Valid	
None	Reject	Value	Not Valid	
Create	None	None	Valid	Object exist but no action is needed.
Create	None	Value	Not Valid	
Create	Accept	None	Valid	The terminal accept creation of this object.
Create	Accept	Value	Not Valid	
Create	Reject	None	Not Valid	
Create	Reject	Value	Valid	The terminal reject creation of this object.
Modify	None	None	Valid	Object exist but no action is needed.
Modify	None	Value	Not Valid	
Modify	Accept	None	Valid	The terminal accept modification of this object.
Modify	Accept	Value	Not Valid	
Modify	Reject	None	Not Valid	
Modify	Reject	Value	Valid	The terminal reject modification of this object.
Remove	None	None	Valid	Object exist but no action is needed.
Remove	None	Value	Not Valid	
Remove	Accept	None	Valid	The terminal accept removal of this object.
Remove	Accept	Value	Not Valid	
Remove	Reject	None	Not Valid	
Remove	Reject	Value	Valid	The terminal reject removal of this object.

Table 48: The meaning of the attributes Operation, Response and Cause in a CS

Operation	Response	Cause	Valid	Description
None	None	None	Valid	Object exist but no action is needed.
None	None	Value	Not Valid	
None	Accept	None	Not Valid	
None	Accept	Value	Not Valid	
None	Reject	None	Not Valid	
None	Reject	Value	Not Valid	
Create	None	None	Valid	Object exist but no action is needed.
Create	None	Value	Not Valid	
Create	Accept	None	Valid	The object is created in network and accepted in terminals.
Create	Accept	Value	Not Valid	
Create	Reject	None	Not Valid	
Create	Reject	Value	Valid	The object is not created because of either network och a terminal has rejected it
Modify	None	None	Valid	Object exist but no action is needed.
Modify	None	Value	Not Valid	
Modify	Accept	None	Valid	The object is modifyed in network.
Modify	Accept	Value	Not Valid	
Modify	Reject	None	Not Valid	
Modify	Reject	Value	Valid	The object is not modifyed because of either network och a terminal has rejected the modification
Remove	None	None	Valid	Object exist but no action is needed.
Remove	None	Value	Not Valid	
Remove	Accept	None	Valid	The object is removed in network.
Remove	Accept	Value	Not Valid	
Remove	Reject	None	Not Valid	
Remove	Reject	Value	Valid	The removal of the object is rejected by either the network or a terminal

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 :
H04M 3/56, H04Q 3/00,
H04L 12/18, 12/66

A3

(11) International Publication Number: **WO 00/11886**

(43) International Publication Date: 2 March 2000 (02.03.00)

(21) International Application Number: PCT/SE99/01445

(22) International Filing Date: 24 August 1999 (24.08.99)

(30) Priority Data:

9802840-0	25 August 1998 (25.08.98)	SE
9902683-3	13 July 1999 (13.07.99)	SE

(81) Designated States: EE, LT, LV, NO, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(88) Date of publication of the international search report:
2 June 2000 (02.06.00)

(71) Applicant: TELIA AB (publ) [SE/SE]; Mårbackagatan 11, S-123 86 Farsta (SE).

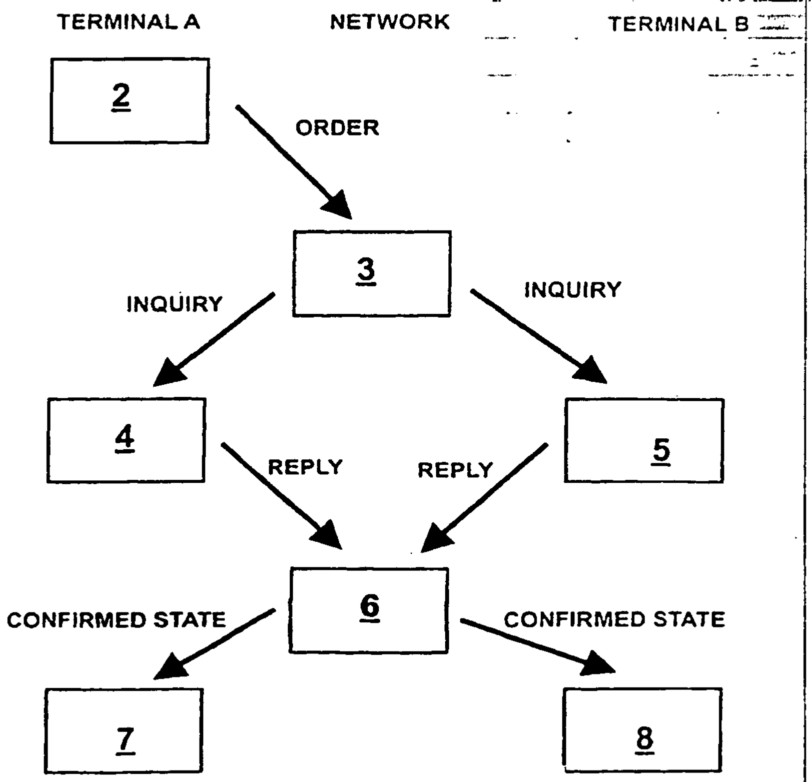
(72) Inventors: BJÖRKMAN, Nils; Fatburs Kvarngata 7, S-118 64 Stockholm (SE). CSEIÉNYI, Istvan; Tuillingarnasgata 317,1, S-136 63 Haninge (SE). ELLDIN, Jonas; Ekehjelmstorget 4,2, S-118 54 Stockholm (SE). GISGÅRD, Christer; Ösmogatan 7, S-122 48 Enskede (SE). LA-TOUR-HENNER, Alexander; Ormingeringen 55 C, S-132 33 Saltsjö-Boo (SE). LUNDBERG, Torbjörn; Finnbergsvägen 18, S-131 31 Nacka (SE). ÖSUND, Stefan; Envägen 15, S-138 35 Älta (SE).

(74) Agent: PRAGSTEN, Rolf; Telia Research AB, Vitsandsgatan 9, S-123 86 Farsta (SE).

(54) Title: IMPROVEMENTS IN, OR RELATING TO, TELESERVICE MANAGEMENT SYSTEMS

(57) Abstract

A teleservice management system is adapted to support the provision of a plurality of complex teleservices and has a plurality of intercommunicating subsystems. The teleservice management system negotiates and settles agreements between participants to a service session, resource control architectures within terminals and a resource control architecture in a transmission network. The teleservice management system employs a teleservice control protocol for transmitting messages between said subsystems and thereby controlling a teleservice. The teleservice control protocol is arranged to link a network resource manager, service users and terminal resource managers, into a teleservice control process for facilitating delivery of a teleservice which is optimal, in terms of resource usage, and agreed by a service user. The teleservice control protocol is adapted for use with a plurality of different teleservices including: multiparty, multimedia conference services; tele-game services; tele-shopping services; and tele-education services. The teleservice control protocol messages refer to specific parts of an object oriented model of a teleservice session. The teleservice control protocol includes messages exchanged between: a terminal part and a network part of the teleservice management system; the teleservice management system and a service control graphical user interface; the teleservice management system and an application launcher daemon; and the teleservice management system and network resource managers.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 99/01445

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04M 3/56, H04Q 3/00, H04L 12/18, H04L 12/66

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04M, H04Q, H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Björkman, N. et al, "The EMMA multimedia conference service", Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on, This Conference was Held:9-12 Sept. 1997, page 1691 - page1695 vol.3 --	1-40
A	Iera, A. et al, "On the management of end-to-end multimedia and multibearer connections in a personal communications system", Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposim on, This Conference was Held: 15-18 Oct. 1996, page 1178 - page 1182 vol.3 --	1-40

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" erlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 28 February 2000	Date of mailing of the international search report 0 7 -03- 2000
Name and mailing address of the ISA Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Henrik Bodin/cs Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 99/01445

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Iera, A. et al, "A Protocol Architecture for Wireless ATM Networks Supporting Multimedia Teleservices", Universal Personal Communications, 1996. Record., 1996 5th IEEE International Conference on, This Conference was Held: 29 Sept.-2 Oct. 1996, page 482 - page 486 -- --------	1-40